Mensile - 25 gennaio 1986 - Anno V - N. 27 - Sped. Abb. Post. Gr. III/70 - CR - Distr. MePe

Protezioni

Tre sistemi per farle e per disfarle

Il joystick senza misteri

Merletti e geometria

Due giochi di strategia

Come profittare dei saldi di gennaio

Sempre più ricca l'enciclopedia delle routine

Systems

100 pagine 100

PRESENTA

Software Club

20 min. Computer Film

C64/C128
Cover
Buddha
Graphic routine
Computer film
CapTron
Merge

C16/+4
Cover
Data Base
Invaders

MSX
Pinguin
Lunar lander

VC20
Cover
War game
Distruction
Invaders

Spectrum
Dancing
Modem
Memorandum
Odissea

In edicola

Commodore Club # 9
Dir. Resp. A. Ronchetti
Edizioni: Systems Editoriale
Srl - V.le Famagosta 75 -
20142 Milano - Reg. Trib. Mi.
n. 104 del 25/2/84 - Distr. MePe

27

Sommario

RUBRICHE

Direttore: Alessandro de Simone
Redazione/collaboratori: Giovanni Bellù, Simone Bettola, Andrea e Alberto Boriani, Giancarlo Castanga, Eugenio Coppari, Maurizio Dell'Abate, Marco De Martino, Luca Galluzzi, Giancarlo Mariani, Flavio Molinari, Enrico Scelsa, D. Matturro, M.L. Nitti, Massimo Patti, Massimo Pollutri, Carla Rampi, Fabio Sorgato, Giovanni Verrelli, Antonio Visconti.
Segreteria di redazione: Maura Ceccaroli, Piera Perin
Ufficio Grafico: Mary Benvenuto, Arturo Ciaglia, Paolo Vertuccio
Direzione, redazione, pubblicità: V.le Famagosta, 75 - 20142 Milano - Tel. 02/8467348
Pubblicità: Milano: Michela Prandini, Giorgio Ruffoni, Roberto Sghirinzetti, Claudio Tidone - Segretaria: Lilliana Degiorgi
● Roma: Spazio Nuovo - via P. Foscari 70 - 00139 Roma - Tel. 06/8109679
Abbonamenti: Marina Vantini
Tariffe: prezzo per copia L. 3.000. Abbonamento annuo (11 fascicoli) L. 28.000. Estero: il doppio. Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 55.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizioni: Systems Editoriale Srl
Fotolito: Systems Editoriale Srl
Stampa: La Litografica S.r.l. - Busto Arsizio (VA)
Registrazione: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70%
Distribuzione: MePe, via G. Carcano 32 - Milano

l'argomento del mese

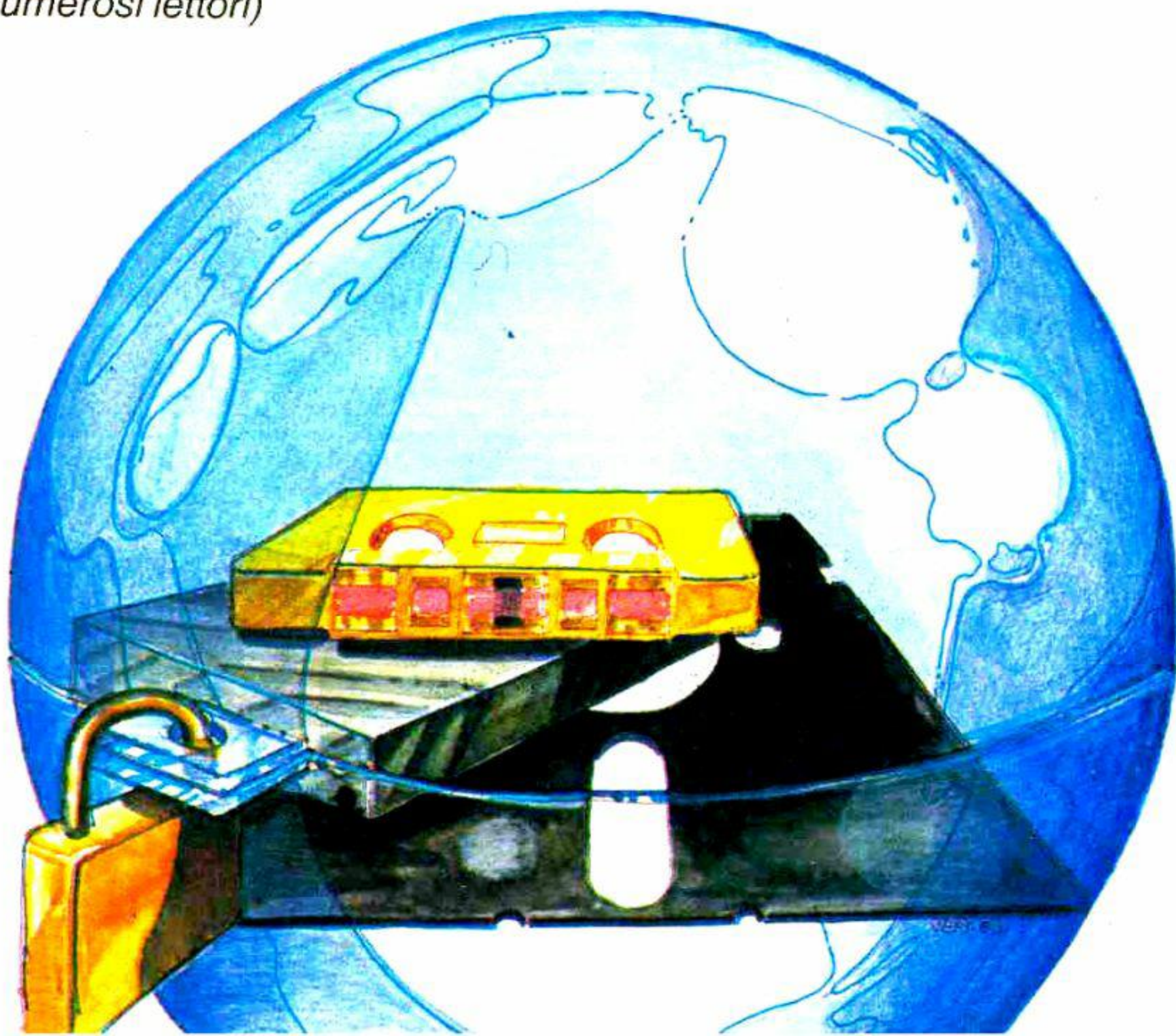
La protezione del software

Perchè non parlate mai di come si fa a proteggere un programma oppure di come eliminare la protezione da quelli normalmente commercializzati? (numerosi lettori)

Entriamo subito nell'argomento suddividendolo in paragrafi che rappresentano, ciascuno, una risposta alle domande più comuni che sono pervenute, numerose, negli ultimi tempi.

I computer dedicati

Per fare in modo che un particolare programma non venga sfruttato commercialmente da altri, esistono in commercio vere e proprie "stazioni di lavoro" capaci di svolgere un solo e ben determinato compito. Si pensi ai sistemi di video-scrittura adoperati nelle redazioni dei giornali o ai calcolatori in grado di svolgere compiti particolarissimi come i controlli di processo nelle produzioni in serie di manufatti di vario tipo.

I programmi capaci di farli funzionare risiedono spesso su ROM e il software ivi residente (denominato, più propriamente, FIRMWARE) si rivela utile e funzionale solo se utilizzato insieme con la circuiteria elettronica "di contorno" di cui fa parte. Se i progettisti lo desiderano, è sufficiente, in questi casi,

utilizzare un monitor che non segua particolari specifiche o collegare una tastiera o una stampante diversa da quella fornita dalla ditta che realizza il sistema per renderlo totalmente inefficiente. In casi come questi duplicare i programmi residenti su ROM non serve assolutamente a nulla se l'elettronica di contorno non è esattamente come previsto.

Riferendosi al campo dell'utenza domestica, le vecchie consolle di videogiochi, che richiedevano cartucce ROM per funzionare, erano sufficientemente protette dal fenomeno della riproduzione indesiderata di programmi perchè, non essendo previsto il caricamento da supporto magnetico, l'unico modo di duplicare un programma era quello di copiarlo ancora su ROM. I costi totali, però, risultavano tali che il prezzo finale all'utente sarebbe stato di poco diverso da quello "ufficiale" proposto dalla Casa madre.

Al contrario, la notevole diffusione di computer dotati di memoria RAM, in cui è possibile caricare, cancellare, duplicare, modificare e creare un qualsiasi programma, ha di fatto reso obsolete le consolle dedicate o funzionanti soltanto con programmi su cartuccia ROM.

Il sistema delle tastiere dedicate, opportunamente riveduto e corretto, potrebbe, a pensarci bene, costituire una valida barriera alla pirateria. Un esempio per tutti può essere rappresentato dal Commodore Plus-4 che, contenendo al suo interno ben quattro programmi di uso universale, rende di fatto completamente autonomo il calcolatore e non richiede l'acquisto di altri package di programmi.

Assisteremo forse alla produzione di una tastiera che funziona solo da (potente) word-processor, un'altra da spreadsheet e così via? La moderna tecnologia, con i circuiti integrati customizzati, può infatti offrire altissima qualità a prezzi stracciati. Una tastiera dedicata solo al word-processor, che sia venduta almeno in un milione di esemplari, potrebbe venire a costare forse meno del solo dischetto-programma di word-processor posto in vendita per un sistema computerizzato MS/DOS!

La protezione personale

Come fare per impedire che nessuno, all'infuori di noi stessi, possa utilizzare un certo programma frutto di laboriose fatiche?

Questo tipo di protezione è la più semplice da applicare, perchè solo l'autore del programma è a conoscenza dei compiti che svolge il programma stesso.

D'ora in poi ci riferiremo, negli esempi citati, a listati brevissimi e facilmente comprensibili anche da principianti. E' ovvio che nessuno si sognerebbe di proteggere programmi così banali e quanto diremo sarà utile esclusivamente per seguire il discorso.

Supponiamo, dunque, che il programma in oggetto sia in grado di calcolare la media aritmetica di 10 numeri da digitare in successione. Una sua versione può essere, tra le tante, la seguente:

```
100 PRINT"CALCOLO DELLA MEDIA"
110 X=0: FOR I =1 TO 10
120 PRINT I;"NUMERO";
130 INPUT Y: X=X+Y
140 NEXT
150 PRINT"LA MEDIA E'"
160 PRINT X/10
```

Una persona non autorizzata che si impossessasse di una copia del programma non avrebbe difficoltà, dopo qualche minuto trascorso alla tastiera, a comprenderne l'uso.

Se però cancelliamo le righe 100, 120 e 150, che si riferiscono alla parte "interattiva" (cioè di dialogo tra calcolatore ed utilizzatore), l'eventuale intruso dovrebbe far far girare più volte il programma prima di capire a che cosa realmente serve.

Un altro modo di confondere le acque è quello di scrivere un programma in modo tale che non sia la riga a numerazione più bassa quella che deve esser elaborata per prima. Scrivete il programma di prima come segue:

```
100 NEXT: GOTO 160
110 X=0: FOR It= TO 10
130 INPUT Y: X=X+Y
140 GOTO 100
150 PRINT X/10
```

Se provate a lanciare il programma col consueto RUN, non ottenete altro che un messaggio: NEXT WITHOUT FOR ERROR IN 100.

Il programma, infatti, per funzionare correttamente, necessita del comando:
RUN 110

E' ovvio che è possibile scrivere il programma in modo che la prima riga ad essere elaborata sia qualsiasi e tale da essere conosciuta solo dall'autore del programma.

I più cattivi possono, addirittura, inserire nella prima riga BASIC un comando come:
90 SYS 64738
che, come è noto, provoca il reset del sistema nel Commodore 64.

Questo esaminato è il sistema più sicuro che esista e non c'è verso che persone non autorizzate possano usare proficuamente un vostro listato. Può essere molto efficace nei casi in cui molti utenti si servano dello stesso computer centrale cui è possibile accedere via terminale. Nel modo descritto, e soprattutto ricorrendo al linguaggio macchina (L.M) piuttosto che al BASIC, è possibile fare in modo che solo voi possiate utilizzare un certo programma, perchè solo voi sapete non solo a che serve ma anche come farlo partire.

Purtroppo questo è anche il sistema... più stupido, perchè per protezione di Software non si intende l'uso "esclusivo" del programma, ma la possibilità che, una volta ceduto ad altri, non venga riprodotto e venduto a terzi senza la vostra autorizzazione.

E' fin troppo ovvio che se vendete un listato non potete evitare di dire come funziona nè tantomeno... che compito è in grado di svolgere!

La protezione del "sorgente"

Volendo impedire, all'utilizzatore del programma, di impossessessarsi di una particolare tecnica di programmazione adottata (alberi binari, soluzione di sistemi di equazioni particolarmente veloce, ordinamento di un vettore, ricerca di un elemento di un arichivio, eccetera) è necessario utilizzare un linguaggio poco conosciuto.

Al giorno d'oggi molte sono le persone che conoscono piuttosto bene il BASIC, ma sono in numero modesto gli esperti di linguaggio macchina, Assembler (strettamente legato al L.M.), Pascal ed altri linguaggi o pseudo-linguaggi. Chi ne è capace, in tal caso, può scrivere il programma direttamente in L.M. e limitarsi a comunicare al cliente soltanto l'esatto indirizzo della SYS di partenza.

Chi non è in grado di scrivere un programma in L.M. può ugualmente fornire all'utilizzatore una versione compilata del programma BASIC originale che risulta altrettanto "illeggibile" di un listato in L.M. Che cosa sia un compilatore lo vedremo tra breve.

Un programma scritto in L.M., o compilato, impedisce a chi lo utilizza di impadronirsi di una particolare tecnica di programmazione (pensate alla generazione degli incredibili effetti sonori o, addirittura, alla riproduzione di parole sintetizzate), ma non impedisce la duplicazione del programma stesso. Moltissimi appassionati di home computer, infatti, posseggono versioni copiate di fantastici videogiochi e, infischiandosene di sapere come "funzionano", si limitano solo a giocare!

La protezione apportata dal codice macchina non risolve, nemmeno in questo caso, la piaga della pirateria sofware.

Le protezione anticopia

E, finalmente, affrontiamo il problema nell'aspetto più interessante e richiesto.

E' possibile proteggere validamente un programma da copie non autorizzate? La risposta non può essere che laconica: no nel modo più assoluto!

Il motivo è piuttosto semplice: dato che c'è un codice, l'unico problema è scoprirlo. E' come una serratura: è sufficiente avere la chiave per accedere all'appartamento.

Se un commerciante disonesto vende serrature di cui conserva copia della chiave oppure antifurti elettronici di cui annota la combinazione, può cedere l'una o l'altra a scassinatori con cui è in combutta perchè la "protezione" sia inefficace.

L'amara vicenda del Bancomat (che ha consentito di contraffare tessere magnetiche che abilitavano al prelievo elettronico di denaro negli appositi sportelli automatici) dimostra che non è assolutamente possibile prevenire con certezza azioni illegali o, comunque, indesiderate.

L'unico modo per scoraggiare la pirateria software, a mio avviso, è solo quello di seguire una seria politica commerciale che impedisca di fatto, con un prezzo oltremodo contenuto, la duplicazione dei programmi.

Un listato che, per il Commodore 64, viene posto in vendita a trecentomila lire (cifra quasi eguale al computer su cui gira) rappresenta, praticamente, una sfida aperta: gli sprotettori, acquistata una copia "regolare", eliminano la protezione per puro e semplice diverti-

mento, sostenuti, alla fine, dalla soddisfazione di aver sprotetto (e divulgato, magari gratuitamente) un programma che, al pubblico, costa una cifra consistente.

Se lo stesso programma fosse venduto a ventimila lire, in confezione elegante, con dettagliate e chiare istruzioni in italiano, molto probabilmente non subirebbe il fenomeno della duplicazione incontrollata, proprio perchè, tra supporto magnetico e fotocopie del manuale, l'impresa non varrebbe la spesa.

Le protezioni più diffuse

Per evitare una copia indesiderata si ricorre quasi sempre a particolari procedimenti che, una volta caricato il programma, lo fanno immediatamente partire. Contemporaneamente viene inibita la funzione dei tasti Run/stop e Restore e non è possibile arrestare l'esecuzione del programma o effettuarne la copia. Il problema, a questo punto, si presenta distinto a seconda se il programma protetto sia su cassetta oppure su disco.

Nel primo caso, il programma può essere duplicato nella maniera più semplice che esista. Ricorrendo ad uno dei duplicatori, la cui pubblicità appare spesso su Commodore Computer Club, è possibile riprodurre qualsiasi nastro, sia esso redatto in BASIC o compilato, scritto in L.M. oppure registrato col Turbo Tape.

Quegli apparecchietti diabolici, infatti, non fanno altro che prelevare parte del segnale elettronico durante la fase di lettura e trasferirlo, semplicemente, su un secondo registratore (originale oppure no) predisposto per la registrazione. Alla fine dell'operazione, nel secondo registratore sarà presente una copia perfettamente identica all'originale, protezioni, trucchi e trucchetti compresi!

E' pertanto inutile dedicare tempo per individuare una protezione da applicare a programmi su nastro.

L'uso del drive per dischetti, invece, consente protezioni più efficaci. Il motivo risiede nel fatto che è possibile alterare, in parte, il sistema operativo del drive stesso generando, ad esempio, volontari errori di lettura che impediscono, di fatto, una copia del disco in tutto o in parte.

Tentando di effettuare un Back-Up (copia intera del disco) il drive copiatore cade inesorabilmente in errore e non porta a termine l'operazione. Oppure sembra che l'operazione sia avvenuta regolarmente ma, tentando di utilizzare il disco-copia, il programma non "parte".

Approfondire il discorso sulle tecniche di protezione su disco non fa parte degli scopi del presente articolo che si ripromette, al contrario, di dare solo alcuni consigli su come proteggere il proprio software e, per di più, solo in casi particolari.

I ragionamenti che seguiranno, infatti, partono dal presupposto che non è assolutamente possibile, nè con la tecnologia attuale nè con quella futura, generare un sistema di protezione efficace!!!

Nel vico del discorso

Vediamo, dunque, di chiarire in quali casi è possibile applicare una qualche protezione al software.

Escludiamo, a priori, la possibilità di usare un registratore, dato che il numero di utenti Commodore che posseggono un duplicatore è piuttosto elevato.

Riferiamoci, pertanto, solo a sistemi che utilizzano il drive.

Dobbiamo, tra l'altro, escludere o limitare software di larghissima diffusione (videogiochi) ed il perchè lo capirete tra breve.

Potrà proteggere il proprio software, forse anche per vie legali, solo chi si dedica a stendere programmi professionali "su misura".

I consigli che seguiranno, in parte applicati personalmente con un certo successo e in parte noti per "sentito dire", non sono altro che un invito ai lettori a proporre qualcosa di analogo che abbia una considerevole possibilità di creare una "protezione".

E' infatti inutile (e ipocrita) descrivere in dettaglio una certa tecnica perchè, una volta pubblicato un articolo su come utilizzarla, automaticamente, tra le righe, viene suggerito il modo di sproteggere. Sarebbe come pubblicare il codice segreto per entrare in contatto con la banca dati della NASA.

La continuità dell'assistenza

Un modo di evitare la copiatura dei programmi è quello di privilegiare committenze di programmi che richiedono frequenti aggiornamenti.

Un programma di contabilità che debba tener conto di nuove leggi, decreti, nuove tabelle ed altro, offre di certo più garanzie di un programma "universale" come un word-processor (o videogioco), che non richiede "manutenzione" di alcun tipo.

Non appena si vara una legge che impone variazioni, il cliente è inevitabilmente costretto a chiamarvi per le opportune modifiche al programma da voi fornito in precedenza. In questi casi, che possono esser più frequenti di quanto si pensi, il cliente scorretto difficilmente passerà tutte le volte la nuova versione ad estranei e questi, in molti casi, desisteranno dal richiedere ogni volta l'aggiornamento.

Considerando, tuttavia, questi casi un modo per scoraggiare la duplicazione è quello di inserire in ogni schermata l'intero nominativo della Ditta cliente e così in ogni copia di qualsiasi output su stampante.

In questo caso, infatti, se il titolare della ditta Rossi cede sottobanco una copia del vostro programma alla ditta Bianchi, che figura farà questo titolare con la propria segretaria e con la propria clientela? Immaginate la figuraccia che è costretto a fare nel consegnare ai propri clienti output di stampante su cui figura, cancellato, il nominativo della ditta Rossi coperto, a malapena, dal timbro della ditta Bianchi!

E' probabile, inoltre, che la ditta Bianchi si rivolgerebbe direttamente a voi, se sapesse chi è il vero autore del programma. Per "coprire" questa eventualità ogni tanto fate in modo che sul video compaia il vostro nominativo, con tanto di numero di telefono.

Si potrebbe obiettare che, in un programma del genere, è sufficiente scorrere il listato in modo da rintracciare sia il primo che il secondo "trucco" descritti e cancellare o alterare le linee BASIC che contengono le istruzioni di... scoraggiamento.

Quello che infatti non vi abbiamo ancora detto è che il programma deve, in ogni caso, esser compilato.

Che cosa è un compilatore

Un compilatore è un particolare programma che si incarica di leggere un qualsiasi programma scritto in BASIC e di "tradurlo" in una versione molto vicina al Linguaggio Macchina. Il vantaggio di tale procedimento risiede nel fatto che la velocità di esecuzione diventa decisamente più elevata.

Un programma compilatore, di cui esistono diverse versioni (solo su disco), richiede una procedura semplicissima.

Supponiamo che vogliate compilare un programma BASIC che, ovviamente, avete verificato più volte e non contiene errori di sorta. Questo programma lo registrate (e verificate) sullo stesso disco che contiene il compilatore. Quest'ultimo, caricato e "lanciato", chiede solo il nome del programma che intendete compilare. Dopo aver digitato il nome esatto del file-programma, il compilatore si mette al lavoro che termina in un tempo proporzionale alla lunghezza del programma stesso e che non è mai inferiore ai cinque minuti. Alla fine di tutte le operazioni, sul disco è presente un file-programma (con un nome secondo lo standard specificato sul libretto di istruzioni) che va caricato (LOAD) e lanciato (RUN) come un qualsiasi programma BASIC ma che sembra composto da un'unica istruzione: SYS.

Il programma compilato, benchè sia stato generato su disco, può essere registrato su nastro magnetico e sottoposto a tutte le operazioni cui normalmente sottoponete un qualsiasi programma BASIC ad eccezione della modifica, cancellazione o aggiunta di nuove istruzioni.

I primi "trucchi"

Il vantaggio di consegnare al cliente una copia compilata del programma fa sì che lo stesso listato sia illegibile, almeno da una persona non molto esperta, e, di conseguenza, non modificabile: tutte le intestazioni, scritte, videate, nomi eccetera verranno riportati su video o su stampante senza alcuna possibilità di modifica.

Un altro "trucco" per evitare la duplicazione indesiderata del programma è quello di garantire un certo servizio, magari gratuito, di riordino degli archivi eventualmente generati dal programma.

Nelle varie opzioni del package, ad esem-

pio, fate in modo che, dopo l'inserimento di un certo numero di nominativi, sia necessario un riordino degli stessi, altrimenti, con l'aumentare delle dimensioni dello stesso archivio, la ricerca di un dato avviene dopo un tempo sempre più lungo. Si potrebbe, ovviamente, inserire all'interno dello stesso programma un'opzione che permetta il riordino automatico tutte le volte che se ne presenta la necessità; ma, in questo modo, chiunque potrebbe compiere l'operazione, anche chi possiede una copia di "favore" del listato.

Strutturando il programma, invece, in modo che si richieda il vostro intervento almeno sei volte l'anno, vi garantirete da copie abusive. Dal vostro cliente vi presenterete (ripeto: gratuitamente) con un particolare programma, che non cederete per nessun motivo, che si incarica di "mettere a posto" i file generati dal programma. E' ovvio che chi ne possiede una copia illegale non potrà giustificare la richiesta di un vostro intervento!

Un altro trucco, sperimentato personalmente con successo, consiste nel "legare" il programma con i file che genera. Con la scusa di illustrare all'impiegato (che utilizzerà il computer) i modi di funzionamento, passate un intero pomeriggio (ne vale la pena) e seguitelo almeno nell'inserimento di un centinaio di nominativi. In seguito fate in modo, senza che se ne accorga, che il programma funzioni correttamente solo se sul disco sono presenti, ad esempio, tre nominativi prima digitati e che, ripetuti in una parte, inaccessibile dall'operatore e dal programma, fungano da "chiave" per riconoscere se il programma è usato da persone autorizzate oppure no.

Il programma, in altre parole, funziona solo se in presenza dell'archivio (già iniziato) del cliente. Una copia del programma funzionerebbe solo se si fornisse anche l'archivio che ha generato (tra cui i tre nominativi "chiave"). Non può esser commercializzato un programma che funziona con un archivio riempito in parte!

Se l'utente richiede l'opzione di cancellatura di nominativi, voi realizzerete una particolare opzione che si limiterà a non considerare ulteriormente il nominativo indicato. Questo, invece, resterà fisicamente sul disco anche se non più accessibile da programma. Il vostro interevento periodico consentirà, in un secondo momento, di cancellare realmente i nominativi indesiderati grazie al solito programma Top Secret da utilizzare in casi come questi.

Sul prossimo numero, oltre a riprendere il discorso, parleremo di altri trucchetti che hanno consentito, a numerosi programmatori, di proteggere il frutto del proprio lavoro. Ci farà piacere riportare, come abbiamo già detto all'inizio, l'esperienza dei lettori che abbiano affrontato e risolto brillantemente (o meno!) l'annoso problema.

Alessandro de Simone

Mancanza di compatibilità

☐ **Ho notato che alcuni programmi registrati su nastro adatti per il VIC 20, pur non avendo PEEK o POKE, non possono essere caricati su C64. C'è qualche modo per eseguire questa operazione ?** (Rosario Melilli - Chieri)

• Come abbiamo più volte detto, vi sono notevoli differenze tra la struttura della memoria del VIC 20 e del C64. Può accadere che alcuni programmi scritti per VIC 20 possano essere caricati su C64, ma per farlo occorre prima salvarli su disco e da quest'ultimo caricarli su C64. Occorre però sottolineare che solo i più semplici programmi BASIC possono girare su ambedue i calcolatori, non certo complessi programmi scritti in Linguaggio Macchina.

Super Turbo

☐ **Sono riuscito ad ottenere un programma chiamato "SUPER TURBO" che gira sul Commodore 64. Poichè io posseggo un PLUS 4, ho caricato il programma su quest' ultimo, ma non c'è alcun modo di farlo funzionare. Potete darmi qualche suggerimento ?** (Pietro Ciaccio - S. Margherita Belice (AG))

• I calcolatori Commodore 64 e PLUS 4 sono piuttosto diversi fra loro e difficilmente un programma scritto per il primo può girare anche sul secondo. Ciò non accade, di norma, quando il programma è molto semplice e scritto in BASIC. Non è questo, purtroppo, il caso del "SUPER TURBO", programma in linguaggio macchina (L.M.) che funziona egregiamente su C64, ma non sul PLUS 4 (oppure C16) proprio a causa della differente struttura dei due calcolatori.

C16

PLUS4, C16 e caratteri di schermo

☐ **Quando collego il mio PLUS4 al televisore, i caratteri dei bordi escono dallo schermo. Come fare per ovviare a questo inconveniente ?** (Massimo Falcione - Vibo Marina)

• I caratteri dello schermo del PLUS4 e del C16 sono visualizzati ricorrendo ad una dimensione più ampia di quelli del C64. Può così succedere che nel televisore domestico i caratteri del bordo escano

dai limiti dello schermo visibile. Chi possiede un monitor non ha, fortunatamente, questo problema.

Per ovviare all'inconveniente descritto, occorre premere il tasto "ESC" e la lettera "R": in tal modo il numero di colonne dello schermo si riduce da 40 a 38, e tutti i caratteri risultano così visibili. Il calcolatore, in pratica, crea, ricorrendo al sistema descritto, una finestra grafica sullo schermo.

E' doveroso ricordare, però, che se un programma è stato scritto in modo da sfruttare lo schermo nel formato di 40x25 caratteri, funzionerà male se si ricorre alla tecnica descritta.

Programmi per C16 e PLUS4

☐ **Posseggo un calcolatore PLUS4 e posso facilmente procurarmi programmi per il C16. E' vero che i due calcolatori sono completamente compatibili tra loro?** (Manuele Cecconello - Gaglianico (VC))

• Il C16 ed il PLUS 4 hanno una struttura della memoria interna ed un sistema operativo molto simili fra loro. Non bisogna dimenticare, però, che il PLUS4 ha una zona delle memorie volatili (RAM) molto più estesa del C16, anche se si inseriscono, su quest'ultimo, espansioni di memoria. E' quindi sempre possibile far girare sul PLUS4 programmi scritti per C16, ma non sempre si verifica il viceversa proprio perchè, in alcuni casi, si fa ricorso a tecniche di programmazione valide solo nel modello PLUS4.

Monitor del C16

☐ **Mi interesserebbe molto leggere i programmi in Linguaggio Macchina (L.M.) immagazzinati nel mio C16, ma non so come fare. Mi potete aiutare ?** (Riccardo de Flora - Torno)

• All'interno del C16, come nel PLUS4, è presente un programma "monitor" che permette di disassemblare e operare su tutti i programmi in linguaggio macchina allocati in memoria. Per metterlo in funzione occorre impartire, appunto, il comando MONITOR. Da quel momento si "entra" in L.M. e si hanno a disposizione parecchi comandi per lavorare in linguaggio macchina o Assembler. Ad esempio, per disassemblare un programma, cioè in pratica per leggerlo secondo il linguaggio della MOS Technology, è sufficiente il comando "D" seguito dall'indirizzo di partenza da esaminare. Per tornare al BASIC, si digita "X" e il tasto Return.

L'intero set di comandi disponibili è stato pubblicato, purtroppo, solo sul manuale di istruzioni del PLUS4. La nostra rivista, per venire incontro alle esigenze dei C16sti, ha riportato sul N.22 (luglio agosto '85) un articolo intitolato, appunto, "Le istruzioni del Monitor C16 e PLUS4".

I lettori che desiderano porre questi quesiti sono pregati di compilare esclusivamente la scheda relativa pubblicata in fondo alla rivista.

Gli arretrati di Commodore Computer Club

Molti lettori chiedono spesso informazioni su come ottenere i fascicoli arretrati della Systems Editoriale.

Per ciò che riguarda la nostra rivista (C.C.C.) precisiamo che i numeri 1 e 2 sono esauriti ed è pertanto inutile richiederli. Altri numeri (3, 4, 7, 10) sono in via di esaurimento ed è opportuno, in caso di richiesta, precisare altri numeri nel caso in cui non risultino disponibili al momento dell'ordinazione.

Un numero arretrato costa L.5000 (compresa la spedizione). Nel caso desideriate ordinare quattro (o più) arretrati, conteggiateli per la cifra di 4000 lire cadauno.

E' altresì possibile richiedere, su nastro, i programmi pubblicati su Commodore Computer Club a partire dal N. 12. Ogni nastro, che contiene tutti i listati pubblicati (ad eccezione di 1 riga), viene venduto al prezzo di L.?????( ).

Anche in questo caso, per ordinazioni superiori a N. 4 nastri, applicate uno sconto di L. 1000 per ciascun nastro desiderato.

In nessun caso, quindi, è necessario addebitare le spese di spedizione alle quali provvede l'Ufficio Arretrati.

IN NESSUN CASO SARANNO EFFETTUATE SPEDIZIONI IN CONTRASSEGNO!!!

Per le ordinazioni, inviare in busta chiusa raccomandata un assegno con l'importo esatto insieme con una breve lettera di accompagnamento indicando CHIARAMENTE il vostro nome, cognome, e indirizzo completo di codice postale.

Indirizzate a:
SYSTEMS EDITORIALE
Servizio arretrati
Commodore Computer Club
Viale Famagosta, 75
20142 MILANO

In alternativa, utilizzate un normale modulo di conto corrente postale intestato a:
C/C 37952207
SYSTEMS EDITORIALE
Viale Famagosta, 75
20142 MILANO

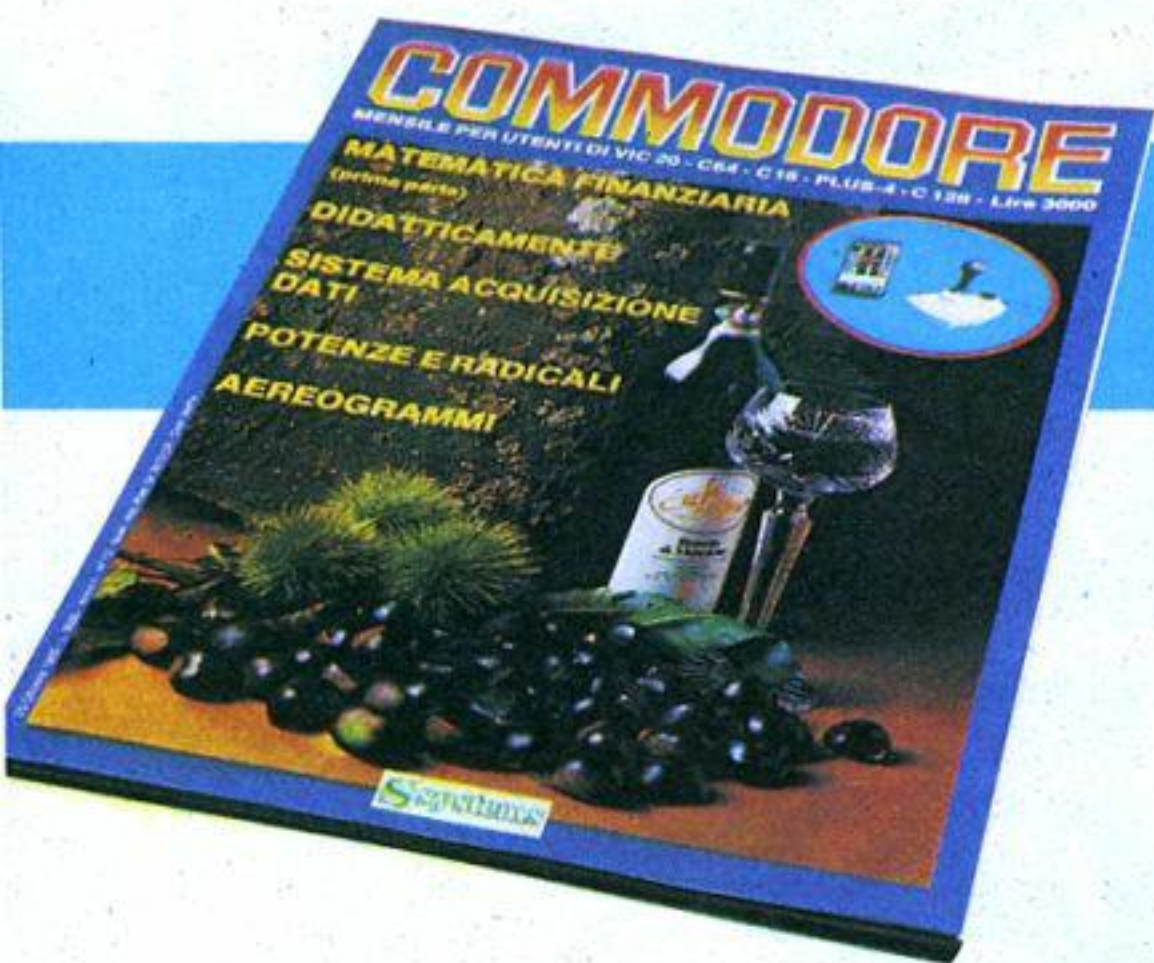
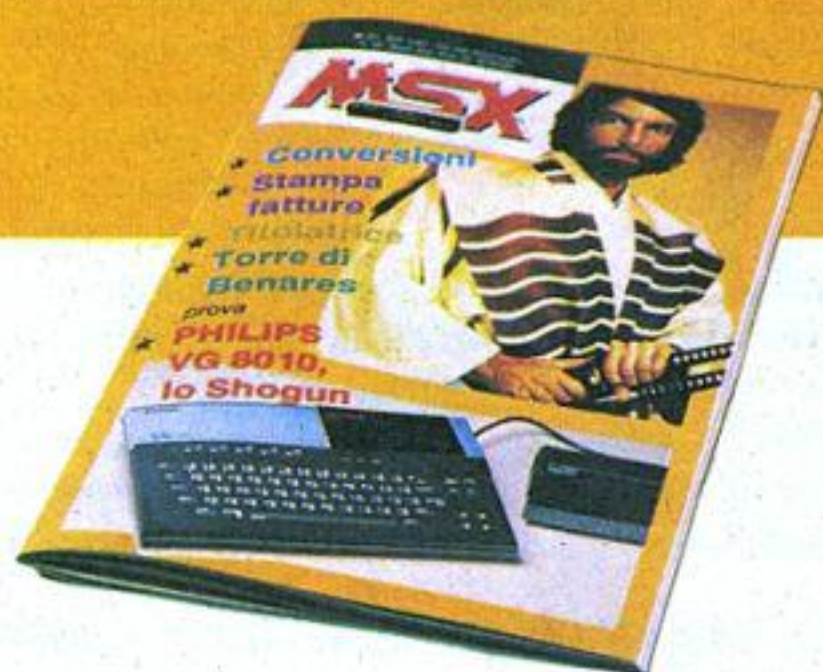
Non dimenticate, in quest'ultimo caso, di segnare sul retro del modulo di conto corrente (nello spazio indicato con la dicitura: CAUSALE DEL VERSAMENTO) sia le vostre richieste che il vostro nominativo completo.

Systems Editoriale: l'aggiornamento continua

Riceviamo molto spesso lettere che ci implorano di pubblicare titoli di libri validi per approfondire questo o quell'argomento.

Molti lettori, infatti, dopo aver preso solenni fregature nell'acquistare volumi che promettevano mari e monti (ma mantenevano laghetti e colline) si sono fatti più prudenti e chiedono maggiori informazioni prima di effettuare altri acquisti.

La Systems Editoriale, esaminando con la massima attenzione le schede informative inviate in gran quantità dai suoi lettori, si presenta sul mercato con un certo numero di testate idonee a soddisfare le più disparate (e disperate) esigenze. In questo fascicolo di Commodore Computer Club le proponiamo soprattutto ai nuovi acquirenti, preoccupandoci, in particolare, di evidenziare a grandi linee le caratteristiche più salienti di ciascuna pubblicazione.

Chi fosse interessato a procurarsi arretrati, oppure a sottoscrivere abbonamenti cumulativi, è pregato di telefonare alla Systems Editoriale (Milano) chiedendo dell'ufficio abbonamenti e arretrati: 02/8467348.

DI RIVISTA.

PUTER

Personal computer

STUDIO DI FUNZIONE

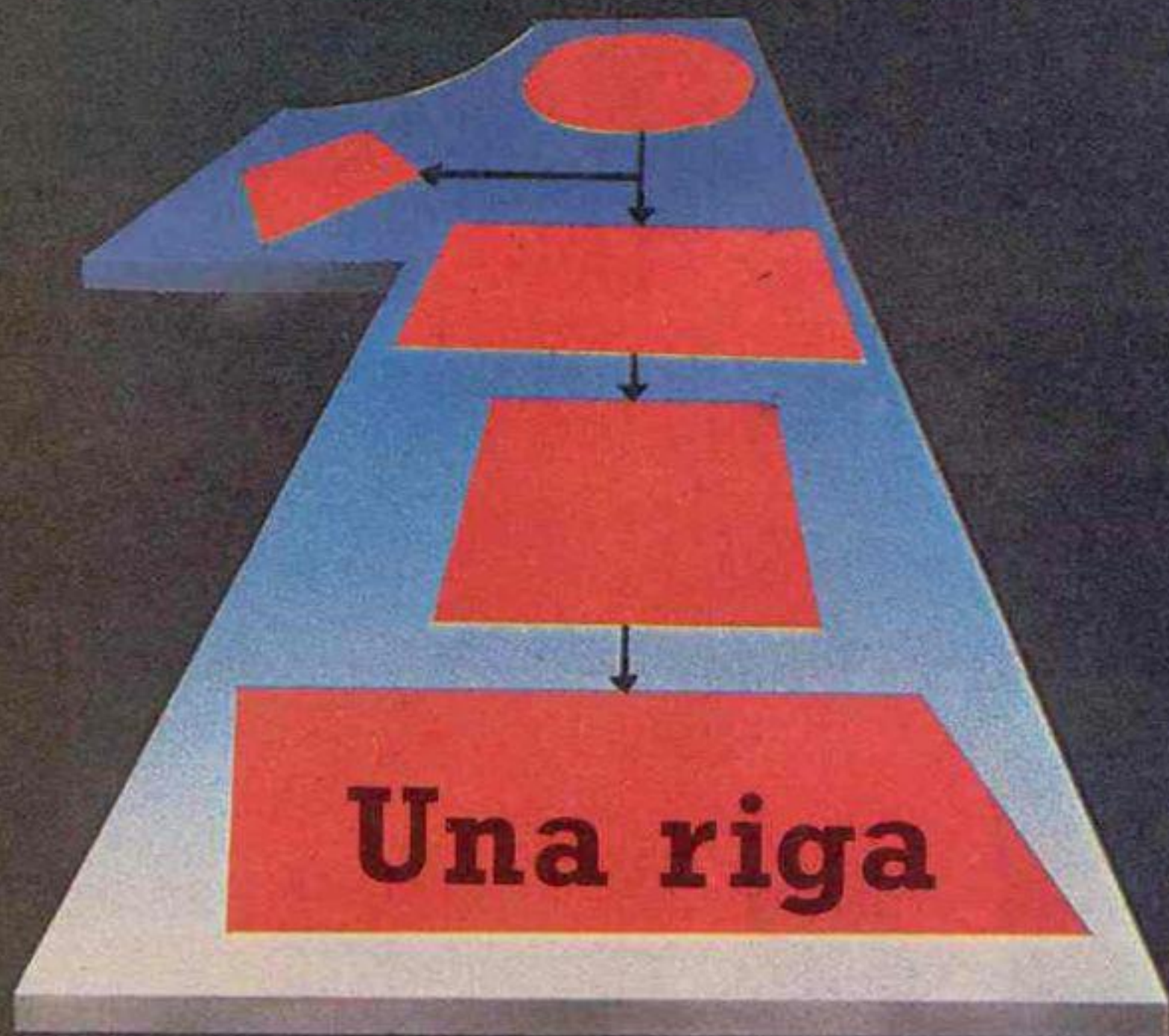
RILOCATORE DI PROGRAMMI

FUNZIONE VAL PER IL QL

TRE RIVISTE IN UNA!

DA GENNAIO £3000

Personal Computer è la nuova rivista Systems per gli utenti Commodore, MSX, Sinclair.
Dal mese di gennaio in edicola, 128 pagine a sole 3000 lire: allo stesso prezzo, la vostra rivista, integra e migliorata, e molto di più.
Non solo tre riviste per tre diversi utenti: **Personal Computer** è anche un'idea nuova per far comunicare tutti gli hobbisti.
Personal Computer: 128 Kbytes di rivista, tutti i mesi in edicola.

Il mercato si evolve. Anche noi.

Una riga

I micro programmi di questa puntata sono dedicati in gran parte (ben 14 Una riga) agli utenti del C16 e Plus4. Negli ultimi tempi, infatti, sono giunte lamentele per la (presunta) mancanza di attenzione, da parte nostra, verso questa categoria "minoritaria".
I listati privi di indicazioni particolari possono girare, come di consueto, su qualsiasi Commodore. Autore dei micro programmi di questo fascicolo è Michele Maggi possessore, beato lui, oltre che di un Plus4, anche di un Commodore 64, due disk drive 1541, stampante MPS 803, plotter 1520 e datassette 1530 e 1531.

1

Radice e potenze (per qualsiasi Commodore). Dato un numero, viene calcolata la relativa radice quadrata, il quadrato e il cubo.

```
1 PRINTCHR$(17):IN
  PUT "NUMERO";A:P
  RINTSQR(A):PRINT
  A↑2:PRINTA↑3:GOT
  O 1
```

2

Rettangoli C16. Un disegno geometrico che viene creato con utilizzando il comando BOX.

```
1 GRAPHIC1,1:
  FORI=1TO130STEP3:
  BOX1,80+I,I+20,200-I,I:
  NEXT:GETKEYA$:
  GRAPHIC0
```

3

Consumi carburanti (per qualsiasi Commodore). Una utilissima una riga per tenere sotto controllo i consumi automobilistici.

```
1 INPUT "LITRI CON
  SUMATI";L:INPUT
  "KM PERCORSI";KM
  :PRINT"KM/L=";KM
  /L:PRINT: RUN
```

4

Area/perimetro (per qualsiasi Commodore). Chieste le misure dei due lati di un rettangolo, ne calcola l'area e il perimetro.

```
1 INPUT "BASE,ALTE
  Z.RETTANGOL";B,H
  :PRINT"AREA";B*H
  :PRINT"PERIMETRO
  ";B*2+H*2:GOTO 1
```

5

Cerchi C16. Un'interessante applicazione del comando CIRCLE per capire come funziona.

```
1 GRAPHIC1,1:
  FORI=2TO90STEP3:
  CIRCLE1,I,I+10,I,I:
  NEXT:GETKEYA$:
  GRAPHIC0
```

6

**Valore CHR$** (per qualsiasi Commodore). Viene chiesto un numero (compreso tra 0 e 255) di cui viene poi visualizzato il corrispondente CHR$.

```
1 INPUT "DIGITA IL
  VALORE DEL CHR$
  ";C:PRINTCHR$(C)
  :GOTO 1
```

7

PRINT musicale C16. Ogni volta che viene premuto un tasto, oltre alla sua visualizzazione sul video, si sentirà un breve impulso sonoro.

```
1 VOL8:GETKEYA$ :
  SOUND2,500,3:
  PRINTA$;:GOTO1:
```

8

Ciambella C16. Una ciambella in Alta Risoluzione che viene disegnata mediante il comando BOX.

```
1 GRAPHIC1,1:
  FORI=1TO180 STEP4:
  BOX1,100,80,220,100,I:
  NEXT:GETKEYA$:
  GRAPHIC0
```

9

Schermo random C16. Una schermata di caratteri casuali che si ripete senza fine.

```
1 SCNCLR:
  FORA=3072TO4071:
  POKEA,(RND(1)*255)+1:
  NEXT:GOTO1
```

10

Random music C16. Una musichetta casuale creata dal C16.

```
1 VOL8:
  A=INT(RND(1)*1000)+1:
  SOUND2,A,7:GOTO1:
```

11

Cambia n. drive (per qualsiasi Commodore e 1541). Serve per portare da 8 a 9 il numero di device del drive 1541. Tutto in una riga sola!

```
1 OPEN 1,8,15,"I0"
  :PRINT#1,"M-W";C
  HR$(119);CHR$(0)
  ;CHR$(2);CHR$(41
  );CHR$(73):CLOSE
  1
```

12

Codici ASCII (per qualsiasi Commodore). Ogni volta che viene premuto un tasto viene visualizzato il valore ASCII corrispondente.

```
1 INPUT "PREMI UN
  TASTO (E IL TAST
  O RETURN)";A$:PR
  INT"ASCII "A$"="
  ASC(A$):GOTO 1
```

#13

Random music II C16. Un'altra musichetta in cui è casuale la frequenza musicale e la sua durata.

```
1 VOL8:
  B=INT(RND(1)*30)+1:
  A=INT(RND(1)*1000)+1:
  SOUND2,A,B:GOTO1
```

#14

Tasti musicali C16. Ogni volta che viene premuto un tasto si udirà un suono in funzione del codice ASCII del tasto premuto.

```
1 VOL8:GETKEYA$:
  SOUND2,ASC(A$)*3,3:
  PRINTA$;:GOTO1:
```

#15

Diapason C16. Questa una riga genera una nota "LA" utile per accordare strumenti a corda (es. chitarra).

```
1 VOL8:
  SOUND2,516,60:
  FORI=1TO2000:
  NEXT:GOTO1:
```

#16

Vortice C16. Un impressionante vortice generato dal vostro C-16 con l'istruzione BOX.

```
1 GRAPHIC1,1:
  FORA=2TO178STEP2:
  BOX1,(160-A),(100-A)
  ,(160+A),(100+A),A:
  NEXT:GETKEYA$:GRAPHI
  C0:
```

#17

PEEK musicali C16. Questa una riga chiede il numero di una locazione di memoria traducendone poi il contenuto in una nota musicale.

```
1 VOL8:
  INPUT"LOCAZIONE";L:
  SOUND2,PEEK(L),5:
  PRINT"PEEK("L")="PE
  EK(L):
  GETKEYA$:GOTO1
```

#18

Takeoff C16. Un effetto sonoro che può essere facilmente utilizzato nei videogiochi di vostra creazione.

```
1 VOL8:
  FORI=0TO999STEP5:
  SOUND1,I,1:NEXT:
  FORI=999TO0STEP-5:
  SOUND1,I,1:NEXT:
  GOTO1
```

#19

Telefono occupato C16. Una simpatica riga che genera il segnale telefonico di occupato.

```
1 VOL8:
  SOUND2,765,7:
  FORT=1TO265:
  NEXT:GOTO1
```

#20

Colori schermo C16. Mostra, per ciascun colore generabile dal C-16, ciascuna delle sette gradazioni.

```
1 SCNCLR:
  FORI=1TO16:
  FORA=1TO7:
  COLOR0,I,A:
  COLOR4,I,A:
  FORT=1TO200:
  NEXTT,A,I
```

Nota Bene

Alcune righe tra quelle pubblicate sembrano possedere più di 80 caratteri e, come tali, inaccettabili dal computer.

Nei casi in cui ci si accorga che la riga è troppo lunga, è necessario ricorrere alle abbreviazioni dei comandi così come indicato nell'appendice specifica riportata nel manuale del computer in vostro possesso.

Ad esempio invece di scrivere PRINT è possibile abbreviare col punto interrogativo (?). Invece di POKE potete scrivere il carattere "P" seguito dal carattere che viene visualizzato premendo contemporaneamente il tasto shift insieme con "O". Tutte le abbreviazioni possibili, lo ripetiamo, sono riportate in una delle appendici di qualsiasi manuale Commodore.

Nel caso sbagliate a digitare i microlistati che superano, in lunghezza, gli ottanta caratteri (SYNTAX ERROR o altri tipi di errore), è necessario, per sicurezza, ribatterli per intero e non apportare modifiche alla riga visualizzata con l'istruzione LIST.

La partecipazione dei lettori è gradita e compensata, in caso di pubblicazione, con materiale della Systems editoriale (libri, fascicoli arretrati, abbonamenti, eccetera).

Inviate le vostre 1 RIGA, purchè nella misura di almeno dieci per volta. Non ci è infatti possibile raggruppare i (moltissimi) micro-programmi che pervengono singolarmente in Redazione.

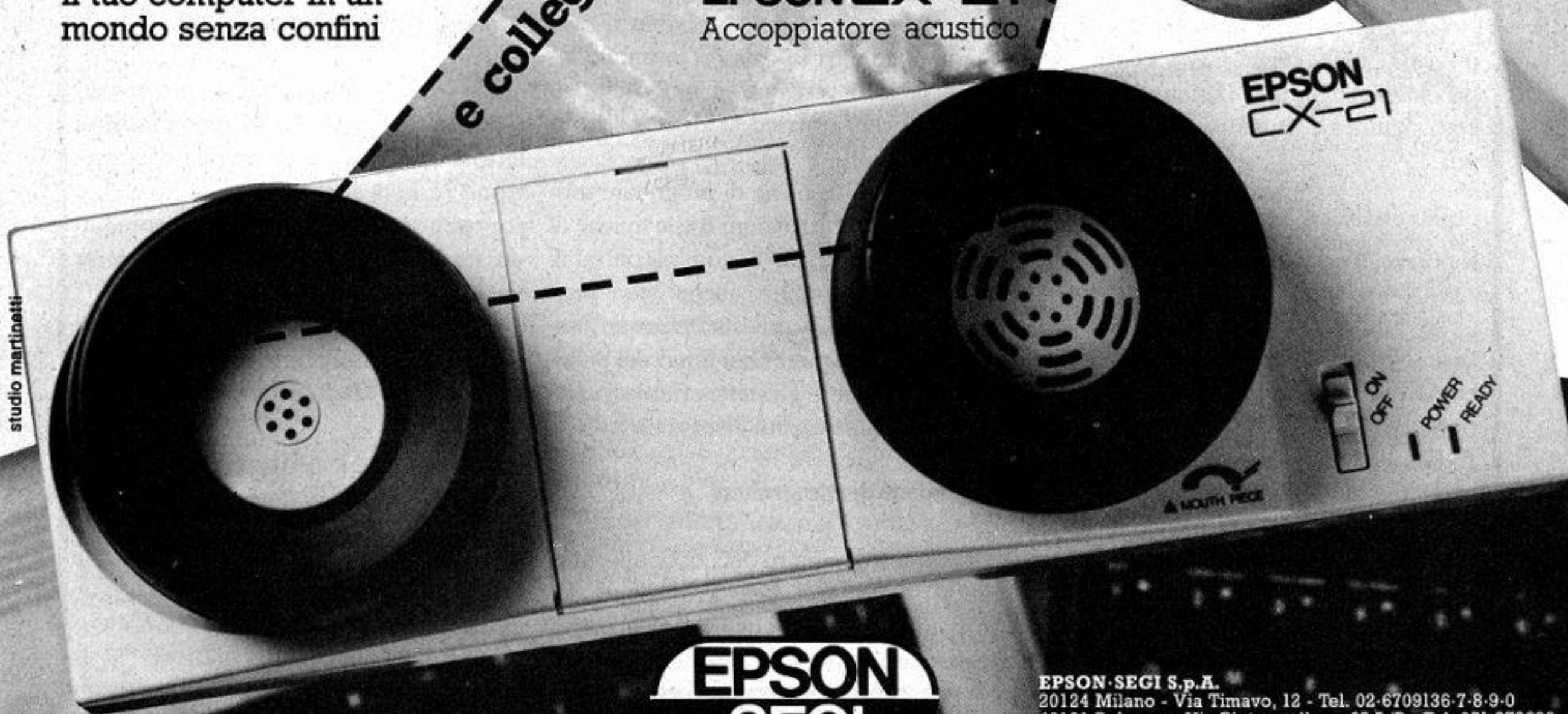
Inviate i vostri lavori su carta, (meglio se su nastro) corredati, ciascuno, di una breve spiegazione sulla funzione che compiono, proprio come li vedete pubblicati in queste pagine.

Ricordate di indicare chiaramente nome, cognome, via e città, e indirizzate a:

Commodore Computer Club
SYSTEMS EDITORIALE

Rubrica "1 Riga"

Viale Famagosta, 75
20142 MILANO

COMPUTERTELEFONA
I TUOI
PROGRAMMI
CX-21 EPSON è un accessorio che dà un valore enorme al tuo calcolatore, quello dei terminali telecomunicazione.
CX-21 è un modem costruito con le tecnologie più avanzate, (modulatore - demodulatore) capace di convertire i segnali digitali in suoni che possono essere trasmessi con un normale telefono, in tutta sicurezza.
Con CX-21 puoi:
- "telefonare" i programmi direttamente al computer dei tuoi amici che posseggono un CX-21
- comunicare con personal computer di altre marche
- collegarti con le banche dati di tutto il mondo
CX-21 è progettato per adattarsi ai diversi apparecchi telefonici per consentire lo scambio veloce e facile di programmi e dati. CX-21 è costruito con speciali circuiti integrati CMOS, è piccolo, leggero e molto affidabile.
Caratteristiche
- Funzionamento a batterie ricaricabili (4 ore di autonomia)
- Funzionamento duplex o half duplex
- Velocità di trasmissione fino a 300 bit-sec.
- Interfaccia RS 232
- Dimensioni 297x95x42 mm
- Peso 700 gr.
Inoltre CX-21 può essere collegato a qualsiasi home o personal computer tramite interfaccia RS 232-C.
e collegati alle banche dati di tutto il mondo
EPSON·SEGI
Garanzia delle Quattro Stagioni
EPSON CX-21,
il tuo computer in un mondo senza confini
EPSON CX-21
Accoppiatore acustico
EPSON
CX-21
ON
OFF
POWER
READY
MOUTH PIECE
studio martinetti
EPSON
SEGI
EPSON·SEGI S.p.A.
20124 Milano - Via Timavo, 12 - Tel. 02-6709136-7-8-9-0
40121 Bologna - Via Pietramellara, 65 A/B - Tel. 051-273686
35128 Padova - Via Pellizzo, 23/9 int. 4/U - Tel. 049-8070870
00199 Roma - Via Asmara, 58 - Tel. 06-8395766

Anno nuovo Systems nuova

Le novità che vedrete fin dall'inizio dell'anno e che, magari, conoscete già.

Avete sicuramente saputo che Commodore Computer Club (la rivista che hai in mano, tanto per intenderci) è da tempo suggerita ai suoi clienti dalla stessa Commodore. Lo dimostra la cedola di abbonamento speciale che potete trovare nelle confezioni Commodore regolarmente distribuite dalla Commodore Italiana. Anzi, se nella confezione che state per comprare non è presente la cedola-sconto, come pure il certificato di garanzia Commodore Italia, potete esser sicuri che la merce proviene dal mercato "parallelo" e non può godere dell'assistenza ufficiale.

Perchè è stata, tra le tante (ma tante davvero) scelta la nostra rivista? Loro dicono che il motivo è da ricercarsi nel numero di copie vendute (il più alto per una rivista che tratta esclusivamente argomenti specifici di una marca di computer) e nella serietà e semplicità dell'impostazione degli articoli.

Noi diciamo, invece, che la Commodore produce apparecchi sufficientemente idonei per supportare il sofware che pubblichiamo. O no?

Scherzi a parte, il fatto che C.C.C. sia indicata, dalla stessa Casa, come fonte privilegiata di notizie Commodore, qualcosa significa e, forsè, anche qualcosa in più.

Il prezzo di copertina

Avrete, ahimè, notato che sono necessarie 500 lire in più per procurarsi una copia della nostra rivista. Non faremo il patetico, tradizionale paragone col prezzo di una tazzina di caffè o con le sigarette. Ci limiteremo a far notare che il corrispondente aumento di pagine è tale da mantenere il rapporto quantità prezzo costante. La qualità, invece, tende al rialzo.

Il motivo dell'aumento del numero di pagine: il numero di lettori che richiedono specifici argomenti è sensibilmente aumentato. D'altra parte eravamo costretti a non trattare temi che, da soli, richiedevano uno spazio eccessivo e non giustificavano la loro presenza in un fascicolo di "sole" 84 pagine. Gli utenti di C-16 e PLUS-4, inoltre, scalpitavano e non si accontentavano di articoli che riguardavano "anche" la loro macchina, ma richiedevano specifiche trattazioni.

La nuova rivista: "Personal Computer"

E' ormai in edicola la nuova creatura della Systems Editoriale, Personal Computer, che si rivolge agli utenti di Commodore, Sinclair, MSX. Per quanto riguarda gli utenti Commodore, la nuova rivista tratterà argomenti un po' più avanzati rispetto agli articoli che potete trovare su C.C.C. Soprattutto, parlerà di quei "legami", apparentemente impossibili, tra le diverse macchine che rappresentano, almeno per ora, la maggior fetta di mercato nel campo degli Small Computer.

Saranno presentati listati di un certo interesse e, soprattutto, si offrirà ai lettori volenterosi la possibilità di collaborare. Se, infatti, saranno pubblicate versioni di listati per un solo determinato computer, i lettori che adatteranno il programma ad altri calcolatori potranno inviare la loro versione in modo che gli utilizzatori di apparecchi identici ne possano usufruire.

Particolari tecniche di programmazione, come pure istruzioni Basic tipiche di una sola macchina e non riscontrabili nelle altre, saranno anche argomenti trattati dalla nuova rivista **Personal Computer** che insegnerà, nei limiti del possibile, ad ottenere gli stessi effetti su macchine completamente diverse tra loro.

Protezione dei programmi

Da questo numero di C.C.C, approfittando dello speciale "Protezione del software", viene inaugurata una nuova "Sezione" completamente dedicata all'interessante argomento. Per stimolare i lettori ad approfondire le proprie cognizioni e a collaborare con C.C.C., la Redazione promuove due iniziative speciali.

Scopri il trucco

Commodore Computer Club e Software Club (la rivista su cassetta della Systems Editoriale) uniscono le loro forze per un nuovo, avventuroso concorso. Per partecipare, basta acquistare un numero di Software Club, ogni mese in edicola, e mettersi al lavoro sul programma che viene presentato protetto (ogni volta con una tecnica diversa).

Al lettore che per primo invierà un breve articolo che illustri il metodo adoperato dalla Redazione per proteggere quel particolare listato, verrà inviato un pacco contenente prodotti della Systems Editoriale.

Si noti che è necessario inviare il metodo usato per proteggere e non quello seguito dal lettore per eliminare la protezione. Siamo infatti sempre contrari a divulgare notizie che possano indirettamente danneggiare l'attività delle Software House.

Proteggi i programmi con C.C.C.

Saranno esaminati e, se meritevoli, pubblicati articoli (brevi, per carità) di lettori che descrivano routine di protezione (e non di sprotezione). Anche, e soprattutto, semplici idee che necessitano studi più approfonditi saranno prese in considerazione e pubblicate.

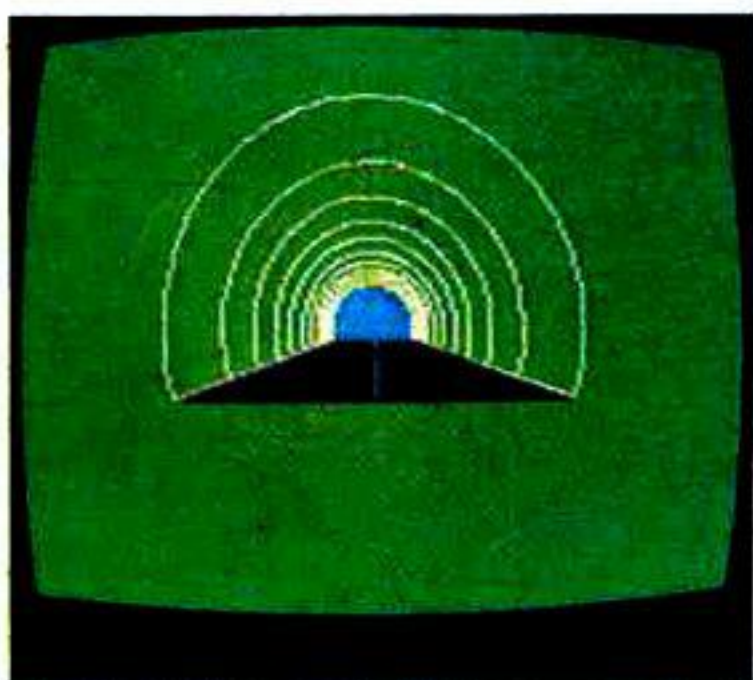
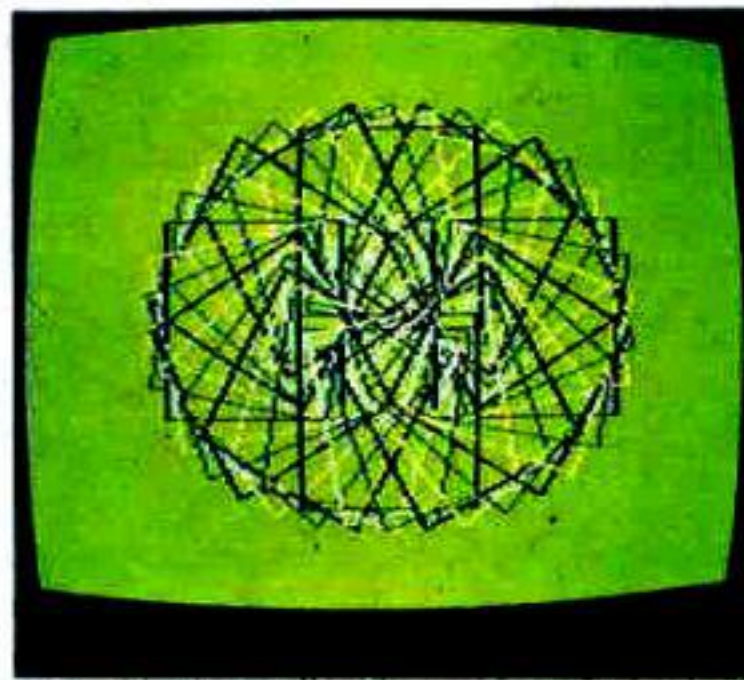
Per collaborare è necessario inviare il materiale, corredato soprattutto del vostro recapito telefonico, a:

Commodore Computer Club
Systems Editoriale
Rubrica "Protezione del Software"
Viale Famagosta, 75
20142 MILANO

COMMODORE 64 C-128

(Solo con le routines grafiche di Danilo Toma)

Figure geometriche

di Riccardo Sclavi

Con questi disegni il C-64 non eguaglia Raffaello, ma si difende bene!

Questi otto disegni in alta risoluzione danno una buona idea delle capacità grafiche del Commodore 64 in alta risoluzione (il termine "hires" che incontrerete nell'articolo non ha nulla si strano: è la contrazione dell'inglese high-resolution, alta risoluzione). Per la sua semplicità non ci sembra il caso di commentare il programma: è costituito da un menu di otto opzioni e da altrettanti sottoprogrammi incaricati di tracciare i grafici.

A proposito di queste subroutines, notate l'assenza di istruzioni del tipo GOTO e GOSUB al loro interno, per cui i più pigri nell'utilizzare i programmi potranno copiare solo le linee contenenti il disegno desiderato.

Sappiate che i disegni iniziano con le linee multiple di 1000 (1000, 2000, 3000 ecc.) e terminano con l'istruzione RETURN, che dovrete sostituire con STOP se volete utilizzare le routines separatamente dal programma.

Attenzione che se l'ultima istruzione è STOP il vostro "cervellone" resta in modo hires e dovete essere voi a riportarlo in modo "normale" (tmodo testo) in uno dei modi seguenti:

- Premete RUN/STOP + RESTORE.

- Digitate alla "cieca" (senza che, cioè possiate vedere ciò che state digitando) il comando TEXT6,1 (preceduto dal carattere di freccia sinistra) seguito dalla pressione del tasto Return.

- Premete un tasto alfabetico seguito da Return, generando però un ?SYNTAX ERROR che riporta, tuttavia, il computer in modo normale.

- Premete SHIFT + CLR/HOME e digitate, sempre alla "cieca", il comando LIST seguito da Return. Quando il programma è listato tutto il calcolatore ritorna al modo normale.

- Digitate il carattere di doppio punto (:) seguito dalla pressione del solito tasto Return.

Detto questo, lascio la parola alle immagini (e alla tastiera su cui dovrete battere prima il programma). Non dimenticate, prima di digitarlo, di caricare ed attivare le routine grafiche di Danilo Toma.

Come procurarsi le routine grafiche

I nuovi lettori di Commodore Computer Club forse non sono a conoscenza del fatto che la nostra rivista ha pubblicato tempo fa (esattamente sul N.14) un programma in linguaggio macchina che aggiunge nuove, potenti istruzioni basic al C-64.

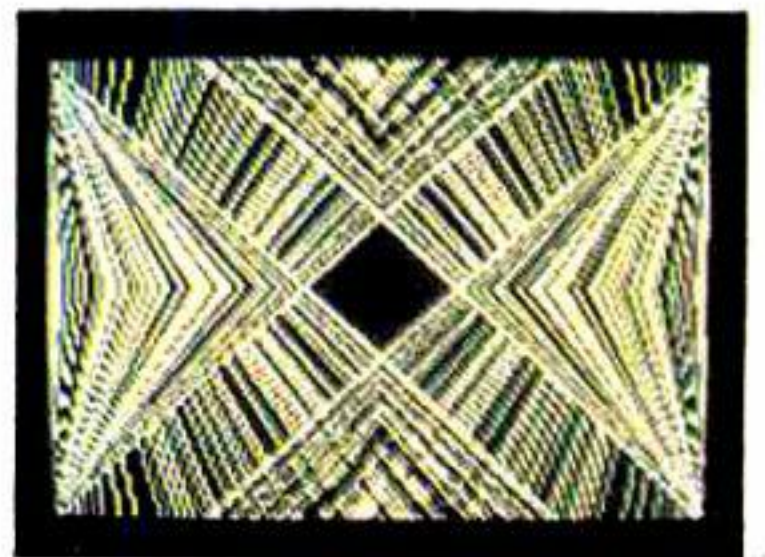
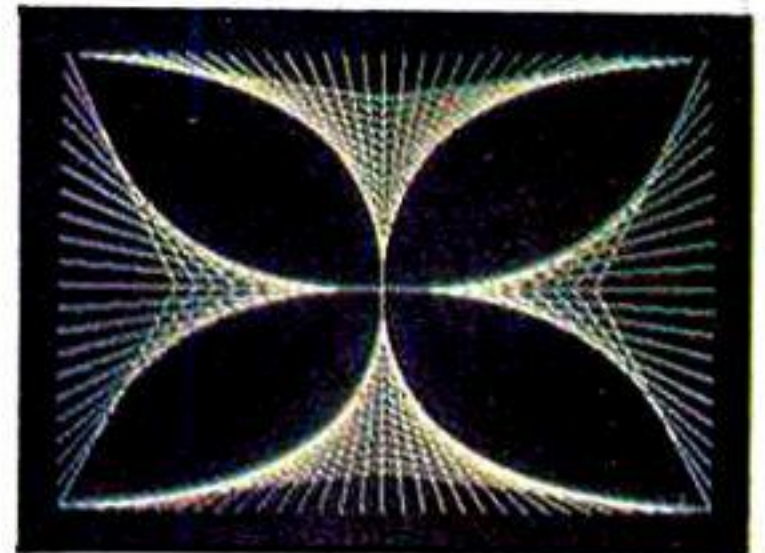
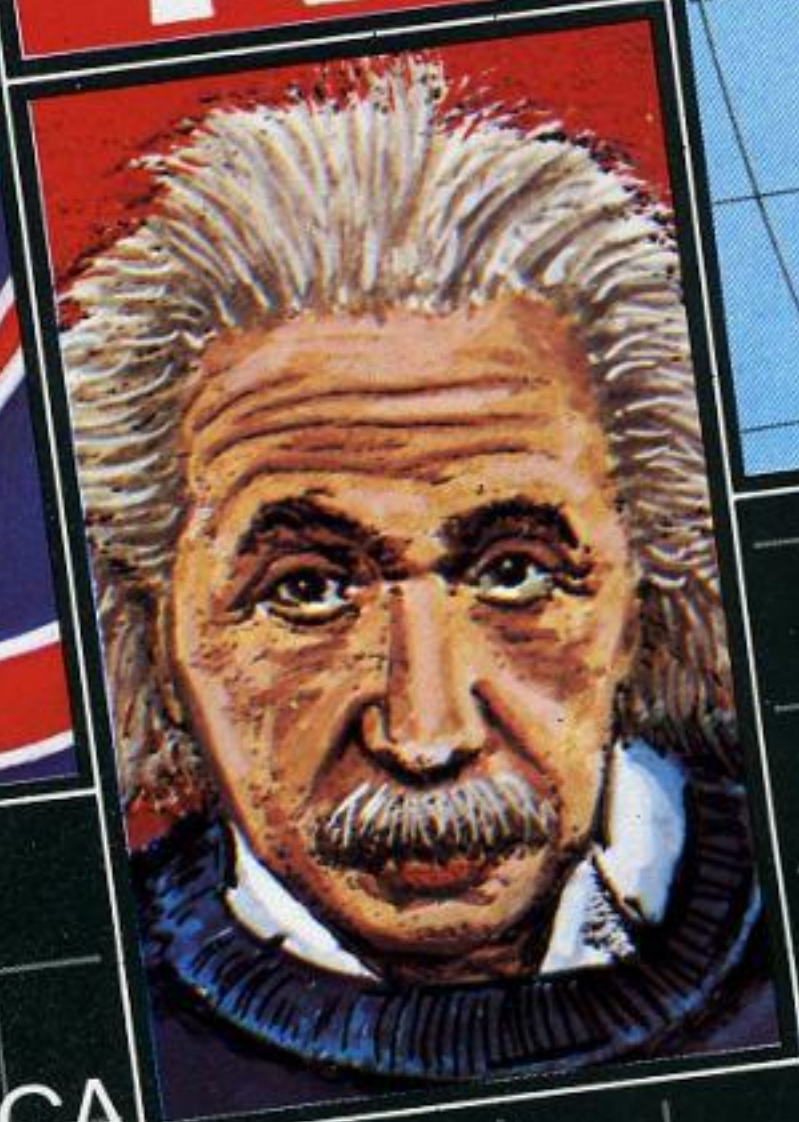
Queste routine, esclusivamente grafiche, consentono di disegnare in tre dimensioni in alta risoluzione.

Digitare il programma è piuttosto lungo e pericoloso e pertanto sconsigliamo, ai lettori inesperti, la sua digitazione.

Per venire incontro alle loro esigenze, tuttavia, la Systems Editoriale invia, su richiesta, in cassetta, le routine grafiche insieme a tutti i programmi pubblicati sul N.14 di C.C.C.

Per avere la cassetta è sufficiente inviare la modica cifra di 5800 (cinquemila ottocento) lire indicando, ovviamente, sia il numero di C.C.C. desiderato sia il vostro nominativo completo. Non si effettuano spedizioni contrassegno.

SYSTEMS EDITORIALE
Servizio arretrati
Viale Famagosta 75
20142 MILANO

```
100 REM COMMODORE 64
110 REM CARICARE LE ROUTINES GRAFICHE
115 REM DEL N.14 DI C.C.C.
116 :
120 REM  * MENU' *
130 PRINTCHR$(147)"DISEGNI 64"CHR$(17):POKE53280,14:←TEXT6,1
140 PRINT"QUADRIFOGLIO -------- 1"
150 PRINT"SPECCHIO ------------ 2"
160 PRINT"GALLERIA ------------ 3"
170 PRINT"ELICA --------------- 4"
180 PRINT"ELLISSE DI QUADRATI - 5"
190 PRINT"OCCHIO -------------- 6"
200 PRINT"CONI 3 DIMENSIONI --- 7"
210 PRINT"SORGERE DEL SOLE ---- 8"
220 PRINT"← ---- RITORNO AL MENU'"
230 GETIN$:IFIN$=""THEN230
240 ONVAL(IN$)GOSUB1000,2000,3000,4000,5000,6000,7000,8000
250 CLR:RESTORE
260 GETPA$:IFPA$<>"←"THEN260
270 GOTO130
1000 REM  * QUADRIFOGLIO *
1010 ←CLEAR:←GRAF0,3:POKE53280,0:←COLOR1
1020 FORX=-160TO160STEP10
1030 ←DRAWX,100,0,0,ABS(X/2),0
1040 ←DRAWX,-100,0,0,-ABS(X/2),0
1050 NEXT
1060 FORY=-100TO100STEP10
1070 ←DRAW-160,Y,0,-ABS(Y),0,0
1080 ←DRAW160,Y,0,ABS(Y),0,0
1090 NEXT
1100 RETURN
2000 REM  * SPECCHIO *
2010 ←CLEAR:←MGRAF0,1,11,15:POKE53280,0
2020 FORN=-160TO90STEP4
2030 ←COLOR4*RND(1)
2040 ←DRAW-160,-100,0,N,100,0
2050 ←DRAW-160,100,0,N,-100,0
2060 ←DRAW160,-100,0,-N,100,0
2070 ←DRAW160,100,0,-N,-100,0
2080 NEXTN
2090 RETURN
3000 REM  * GALLERIA *
3010 POKE53280,5:←CLEAR:←MGRAF5,1,0,14:←COLOR1
3020 FORT=0TO1000STEP100
3030 ←ARC0,0,T,100,100,-.3,π+.3,2*π/40
3040 NEXT
3050 ←COLOR3:FORQ=90TO10STEP-5
3060 ←ARC0,0,1000,Q,Q,0,2*π,2*π/12:NEXT
3070 ←COLOR1
3080 ←DRAW-95,-30,0,-95,-30,1000
3090 ←DRAW95,-30,0,95,-30,1000
```

```
3100 FORP=10TO1000STEP10
3110 ←COLOR2:←DRAW-95,-33,P,95,-33,P:NEXT
3120 ←COLOR1:←DRAW0,-30,0,0,-30,1000
3130 RETURN
4000 REM  * ELICA *
4010 POKE53280,8
4020 ←CLEAR:←GRAF8,7:←COLOR1
4030 FORX=-7.8TO7.9STEP.25
4040 ←DRAWX*20,COS(X)*95,0,(X-.25)*20,COS(X-.25)*95,0
4050 ←DRAW-160,0,0,X*20,COS(X)*95,0
4060 NEXT
4070 RETURN
5000 REM   * ELLISSE DI QUADRATI *
5010 P=π:C%=1
5020 POKE53280,13:←CLEAR:←MGRAF13,0,1,12
5030 FORAN=0TOπ*2STEPπ/18
5040 ←COLORC%:C%=C%+1:IFC%>3THENC%=1
5050 X%=COS(AN)*65
5060 Y%=SIN(AN)*40
5070 ←ARCX%,Y%,0,60,60,π/4+P,9*π/4+P,π/2
5080 P=P+π/18
5090 NEXT
5100 X%=COS(AN)*65
5110 RETURN
6000 REM  * TI VEDO !!! *
6010 ←CLEAR:←GRAF7,6:POKE53280,7:←COLOR1
6015 FORX1=0TOπSTEPπ
6020 FORX=-π/2TOπ/2STEPπ/50
6030 ←DRAW0,0,0,X*100,COS(X+X1)*100,0
6040 NEXTX
6050 NEXTX1
6060 RETURN
7000 REM  * CONI 3D *
7010 ←CLEAR:←GRAF6,7:←COLOR1:POKE53280,6
7020 FORZ1=1200TO0STEP-400
7030 FORX1=-120TO120STEP240
7040 FORAN=0TO2*πSTEP2*π/30
7050 X=COS(AN)*50:Z=SIN(AN)*50
7060 ←DRAW0+X1,100,60+Z1,X+X1,-100,Z+60+Z1
7070 NEXTAN
7080 NEXTX1:NEXTZ1
7090 RETURN
8000 REM  * IL SOLE SORGE ANCORA *
8010 ←CLEAR:←MGRAF0,7,4,1:POKE53280,0
8020 ←COLOR1:FORN=1TO50
8030 ←DRAW0,0,0,(RND(1)*320)-160,RND(1)*90+10,0:NEXT
8040 ←COLOR2:FORX=-160TO160STEP20
8050 ←DRAW0,0,0,X,-100,0:NEXT
8060 FORY=-100TO0STEP10
8070 ←DRAW0,0,0,-160,Y,0:←DRAW0,0,0,160,Y,0:NEXT
8080 RETURN
```

QUALSIASI COMMODORE

Tempo di saldi

di A. de Simone

Molto spesso si sente dire che con la matematica è impossibile unire l'utile al dilettevole.

Il programma pubblicato in queste pagine, valido per qualsiasi computer Commodore, si rivela invece utilissimo sia per imparare con precisione la procedura per l'applicazione dello sconto percentuale, sia per valutare con sicurezza, giocando, la convenienza nell'acquisto della merce offerta in saldo.

Col finire di una stagione i negozianti affiggono, nelle loro vetrine, cartelli che assicurano sconti favolosi. Siete in grado di valutare, a colpo d'occhio, quale è lo sconto realmente applicato?

Come gira il programma

Dopo aver digitato il breve listato, date il RUN. Apparirà, sul vostro schermo, una videata simile alla seguente:.

Indovina lo sconto applicato

Prezzo precedente 77000
Prezzo attuale 73150
Sconto applicato (%)?

Dovrete, ovviamente, digitare la percentuale che ritenete valida. Nel caso particolare, battendo il valore 12, e premendo il tasto Return, il computer risponderà:

Esagerato! Il prezzo, col 12% di sconto, sarebbe: 67760
Sconto applicato (%)?

Digitando, invece, un valore troppo basso, come ad esempio 3, il calcolatore risponde:

E' poco! Il prezzo, col 3% di sconto, sarebbe: 74690

Come avrete intuito la domanda verrà posta fino a che non avrete indovinato lo sconto esatto. In tal caso otterrete come risposta:

Hai indovinato dopo 5 tentativi
Tempo impiegato=000345

Tale risposta, è inutile dirlo, appare se avrete tentato cinque volte nel dare valori diversi, mentre il tempo trascorso dalla formulazione della prima domanda è di zero ore, tre minuti e 45 secondi.

Nel caso non indoviniate dopo dieci tentativi, il computer, impietosito, fornirà egualmente la risposta esatta.

Come funziona il programma

Il listato, talmente breve e commentato, non merita ulteriori spiegazioni.

Ricorderemo, soprattutto ai principianti, che con l'istruzione RND(0) è possibile generare un numero casuale compreso tra "0" e "1".

Provate a far girare il seguente microprogramma:

1 PRINT RND(0): GOTO 1

Vedrete comparire sullo schermo, in rapida successione, una serie infinita di numeri decimali. Poichè raramente può essere utile la generazione di un numero compreso in un intervallo così insolito, è possibile aggirare l'ostacolo moltiplicando lo stesso numero per 10, 100, 1000 e, in definitiva, per una potenza di dieci. Questa operazione, come è noto, "sposta" la virgola di tanti posti quanti ne rappresenta il numero degli zeri della potenza di 10.

Digitate, per chiarire le idee, la seguente riga:

1 PRINT RND(0)*10000: GOTO 1

Potrete notare, in questo caso, che la sequenza di numeri che appare è compresa tra "0" e "9999". Se interessa la sola parte intera dei numeri che compaiono, modificate la riga come segue:

1 PRINT INT(RND(0)*10000):
GOTO 1

Nel programma pubblicato, però, è necessario che il prezzo generato casualmente sia una cifra... credibile. Tutti i prezzi sono, in genere, arrotondati alle 500 oppure 100 lire. Per ottenere quest'ultimo arrotondamento è necessaria un'ultima, semplice modifica:

1 PRINT INT(RND(0)*10000)*100:
GOTO 1

Effettuando la moltiplicazione per 100, tale ultima operazione, aggiunge due zeri "in coda" al numero generato casualmente.

Il lettore, "giocando" con l'istruzione RND(0), può, a sua volta, generare una casistica pressochè infinita di intervalli numerici. L'esperienza acquisita in tal modo potrà essere sfruttata, ad esempio, per scrivere programmi in cui si rivela utile la generazione dei numer casuali (giochi di carte, roulette, adventures, eccetera).

```
100 REM  INDOVINA LO SCONTO
110 REM  PROGRAMMA
120 REM  PER QUALSIASI COMPUTER
130 REM  A.DE SIMONE
140 REM  DIDATTICA '86
150 :
160 PRINTCHR$(147)"INDOVINA LO SCO
    NTO APPLICATO"CHR$(13)
170 X=RND(0):REM  GENERAZIONE NUME
    RO CASUALE COMPRESO TRA "0" E
    "1"
180 A=INT(X*100):REM  "ESTRAZIONE"
     DI NUMERO COMPRESO TRA "0" E
    "100"
190 X=A*100:REM  ULTIME DUE CIFRE=
    00 (PREZZO COMPRESO TRA 0 E 10
    0000
200 Y=RND(0):REM  GENERAZIONE VALO
    RE CASUALE
210 B=INT(Y*40):IF B=0 THEN 200:RE
    M  "ESTRAZIONE" SCONTO PERCENT
    UALE (ESCLUSO "0")
220 REM  VISUALIZZAZIONE MESSAGGI:
     PREZZO PRIMA E DOPO LO SCONTO
230 PRINT:PRINT"PREZZO PRECEDENTE"
    X*10
240 REM  APPLICAZIONE DELLO SCONTO
250 PRINT"PREZZO ATTUALE"INT(X-(X*
    B/100))*10:W=0:REM  W=CONTATOR
    E DI TENTATIVI
260 TI$="000000":REM  AZZERAMENTO
    DEL TEMPO
270 REM  INTERVENTO DEL COMPUTER N
    EL CASO NON SI RIESCA AD INDOV
    INARE
280 IF W>10 THEN PRINTCHR$(18)"TE
    LO DICO IO..."B:GOTO 380
290 INPUT "SCONTO APPLICATO (%)";Z
    :W=W+1:Q=INT(X-X*Z/100)*10
300 REM  CASO DELLO SCONTO INDOVIN
    ATO
310 IF Z-B=0 THEN 360
320 REM  CASO DELLO SCONTO ECCESSI
    VO
330 IF Z-B>0 THEN PRINT"ESAGERATO!
     IL PREZZO, COL"Z"% DI SCONTO
    SAREBBE:"Q:GOTO 280
340 REM  CASO DELLO SCONTO INSUFFI
    CIENTE
350 IF Z-B<0 THEN PRINT"E' POCO! I
    L PREZZO, COL"Z"% DI SCONTO, S
    AREBBE:"Q:GOTO 280
360 PRINTCHR$(18)"HAI INDOVINATO D
    OPO"W" TENTATIVI!"
370 PRINT"TEMPO IMPIEGATO="TI$
380 PRINTCHR$(17)"PER CONTINUARE P
    REMI UN TASTO"
390 GET A$:IF A$="" THEN 390
400  RUN
```

QUALSIASI COMMODORE

La merce più economica

Dopo aver dato il RUN, sullo schermo appare una videata del tipo:

MERCE A. N.ETTI: 3.5 PREZZO L.1400
MERCE B. N.ETTI: 8.5 PREZZO L.3825

QUAL'E' LA PIU' ECONOMICA? (A/B)

Viene, in altre parole, visualizzata una situazione classica che si verifica nei negozi che non si attengono alle disposizioni legislative. A questo punto dovete premere uno dei due tasti (A/B) per comunicare al computer quale delle due merci considerate più economica. Se, nel caso particolare esaminato, digitate "B", otterrete in risposta:

MERCE INDICATA= B
HAI SBAGLIATO
1400 / 3.5 =L. 400 L'ETTO
3825 / 8.5 =L.450 L'ETTO

PREMI UN TASTO

Premendo un tasto qualunque, ovviamente, il computer proporrà un nuovo confronto. Dopo aver trascorso un pò di tempo al computer, meglio se insieme con amici e familiari, avrete sicuramente acquisito una sufficiente esperienza che vi sarà utile, tra l'altro, per individuare quale è la confezione più conveniente di un particolare prodotto (caffè, formaggi, detersivi, eccetera).

Recentemente una disposizione di legge obbliga i commercianti ad indicare il prezzo netto unitario dei prodotti alimentari. Non tutti, però, si attengono a tale disposizione e questo programma può esser utile per individuare, a casa, quali sono i prodotti più economici.

Come funziona il programma

Anche il listato di queste pagine si basa sull'utilizzo intensivo della funzione RND(0). In questo caso, però, è stato necessario selezionare con cura il numero casuale in modo da evitare incongruenze o indovinelli troppo semplici da risolvere.

La differenza del prezzo unitario, infatti, deve esser modesta in modo che sia più difficile indovinare quale delle due merci è più cara dell'altra. Questo problema è stato risolto escludendo tutti i valori unitari che differiscano tra loro di oltre il 20%.

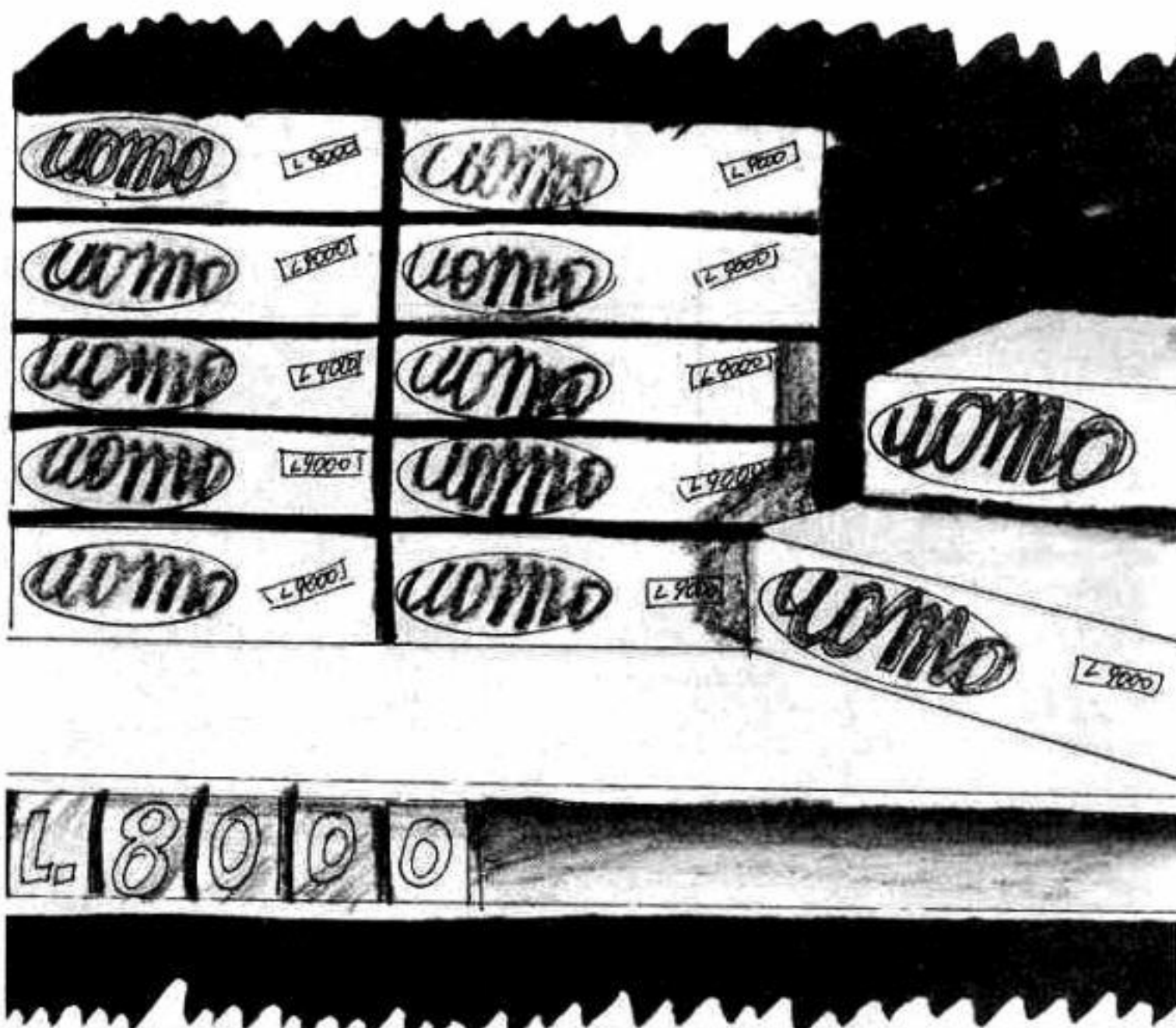
Inoltre i due pesi (e, analogamente, i due prezzi), devono esser diversi tra loro, altrimenti l'indovinello è fin troppo facile.

Esempi dei casi evitati:

Peso 4.7 Prezzo 4300
Peso 8.2 Prezzo 2300

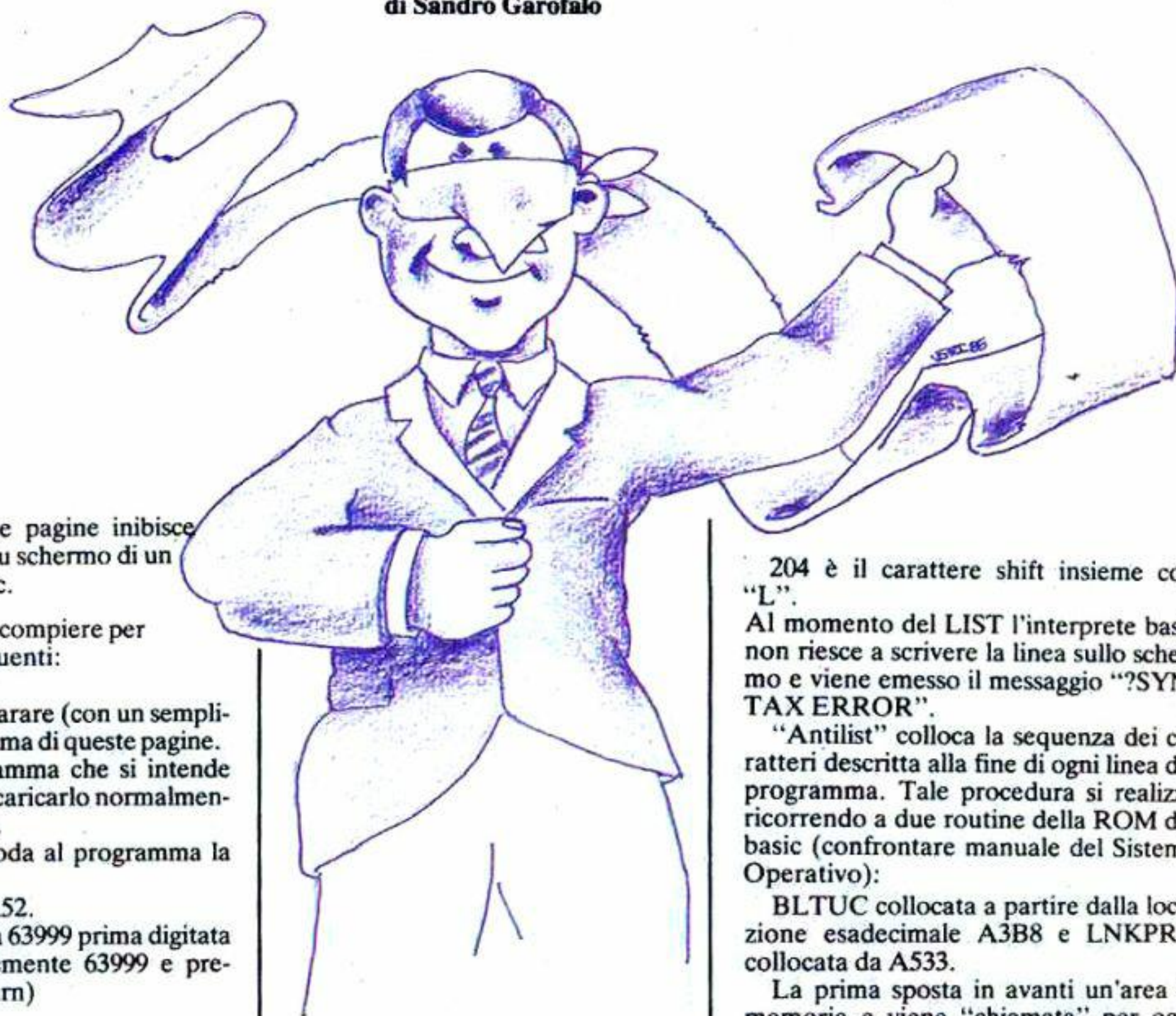
Peso 4.7 Prezzo 4550
Peso 4.7 Prezzo 5750

Peso 5.2 Prezzo 5500
Peso 4.7 Prezzo 5500

• I lettori più volenterosi potranno inserire facilmente ulteriori modifiche, in modo, ad esempio, di trasformarlo in gioco per più giocatori, oppure di individuare quale fra tre confezioni (e non tra due) sia la più conveniente, eccetera.

```
100 REM  QUAL'E' LA
110 REM  MERCE PIU'
120 REM  ECONOMICA?
130 :
140 REM  QUALSIASI COMPUTER
150 :
160 REM  A. DE SIMONE
170 REM  DIDATTICA '86
180 :
190 PRINTCHR$(147)"ATTENDERE..."
200 GOSUB 350:X=X1:REM  PESO MERCE
    A
210 GOSUB 350:Y=X1:IF X=Y THEN 210
    :REM  PESO MERCE B
220 GOSUB 350:XB=X1*100:XA=X*XB:RE
    M  PREZZO MERCE A
230 GOSUB 350:YB=X1*100:YA=Y*YB:RE
    M  PREZZO MERCE B
240 IF XB<YB AND XB/YB<.8 THEN 200
    :REM  ESCLUDE DIVARIO ECCESSIV
    O
250 IF YB<XB AND YB/XB<.8 THEN 200
260 IF YB=XB THEN 200:REM  ESCLUDE
     PREZZI EGUALI
270 PRINTCHR$(147)CHR$(13)"MERCE A
    . N.ETTI:"X" PREZZO L."XA"
280 PRINT"MERCE B. N.ETTI:"Y" PREZ
    ZO L."YA"
290 PRINT:PRINT"QUAL'E' LA PIU' EC
    ONOMICA? (A/B)":TI$="000000"
300 GET A$:IF A$="" THEN PRINTCHR$
    (19)TI$:GOTO 300
310 IF A$="A"  OR A$="B" THEN GOSU
    B 390
320 GOTO 300
330 :
340 REM  GENERAZIONE NUMERI CASUAL
    I
350 X1=INT(RND(0)*10)+.5*(RND(0)>.
    5):IF X1<1.5 THEN 350
360 RETURN
370 :
380 REM  VISUALIZZAZIONE RISULTATO
390 Z1=XA/X:Z2=YA/Y:FOR I=1 TO 5:P
    RINT:NEXT:PRINT"MERCE INDICATA
    = "A$
400 IF Z1<Z2 AND A$="A" THEN 430
410 IF Z2<Z1 AND A$="B" THEN 430
420 PRINT"HAI SBAGLIATO":GOTO 440
430 PRINT"HAI INDOVINATO!"
440 PRINTXA"/"X"=L."XA/X"L'ETTO"
450 PRINTYA"/"Y"=L."YA/Y"L'ETTO"
460 PRINT:PRINT"PREMI UN TASTO"
470 GET X$:IF X$="" THEN 470
480  RUN
```

PROTEZIONI PER I TUOI PROGRAMMI

Protezione antilist

di Sandro Garofalo

Il listato di queste pagine inibisce la visualizzazione su schermo di un qualsiasi listato basic.

Le operazioni da compiere per attivarlo sono le seguenti:

- Caricare e far girarare (con un semplice RUN) il programma di queste pagine.
- Battere il programma che si intende proteggere oppure caricarlo normalmente da nastro o disco.
- Aggiungere in coda al programma la riga: 63999 REM
- Digitare SYS 49152.
- Cancellare la riga 63999 prima digitata (battendo semplicemente 63999 e premendo il tasto Return)

Come funziona

"Antilist" si colloca nella memoria a partire dalla locazione 49152 e manipola direttamente le linee del programma basic.
Il funzionamento è piuttosto semplice: ad ogni linea basic viene aggiunta una REM seguita da alcuni caratteri speciali. La sequenza completa di tali valori numerici è la seguente:

58 143 34 141 147 34 204

Il numero 58 è il codice ASCII del doppio punto (:) separatore, come è noto, di due istruzioni basic.

L'istruzione REM è invece rappresentata dal valore 143.

34 è il codice delle virgolette (").

141 è il carattere speciale che provoca il ritorno a capo.

147 è il codice di cancellazione schermo.

204 è il carattere shift insieme con "L".
Al momento del LIST l'interprete basic non riesce a scrivere la linea sullo schermo e viene emesso il messaggio "?SYNTAX ERROR".

"Antilist" colloca la sequenza dei caratteri descritta alla fine di ogni linea del programma. Tale procedura si realizza ricorrendo a due routine della ROM del basic (confrontare manuale del Sistema Operativo):

BLTUC collocata a partire dalla locazione esadecimale A3B8 e LNKPRG collocata da A533.

La prima sposta in avanti un'area di memoria e viene "chiamata" per ogni linea di programma da proteggere, in modo da creare lo spazio necessario per inserire la sequenza della REM.

La seconda serve a ristabilire i puntatori della catena del basic che, in seguito allo spostamento di memoria e l'inserimento della REM, sono tutti alterati.

Una volta protetto, il programma può essere salvato o fatto girare regolarmente.

E' doveroso ricordare che la routine "Antilist" non è completamente efficace per chi tenta di listare il programma protetto per mezzo di una stampante. In questo caso, infatti, pur se con difficoltà, un utilizzatore sufficientemente esperto riesce a leggerlo.

La routine proposta, tuttavia, è di notevole interesse per la sua brevità e la consigliamo come base di studio per nuove interessanti applicazioni nel campo del Linguaggio Macchina.

Un brevissimo programma in Linguaggio Macchina che può dare del filo da torcere a chi tenta di esaminare i vostri programmi in basic.

```
 10 REM  ANTILIST
 11 REM  PER COMMODORE 64
 12 :
 13 REM  DI SANDRO GAROFALO
 15 REM  GENOVA
 16 :
 17 REM  ROUTINE IN L.M. CHE PROTE
    GGE
 18 REM  DAL LIST I PROGRAMMI IN B
    ASIC
 20 :
100 PRINTCHR$(147) TAB(15)CHR$(18)
    "ANTILIST"
110 C$=CHR$(13):K=0:FOR I=49152 TO
     49269
120 READ A:POKE I,A:K=K+A:NEXT
130 IF K<>14213 THEN PRINT"ERRORE
    NEI DATI":END
140 PRINTC$;C$"CARICARE IL PROGRAM
    MA BASIC DA"
150 PRINT"PROTEGGERE E DIGITARE "C
    HR$(18)"SYS 49152"
300 DATA 173,1,8,133,95,173,2,8,13
    3,96,56,165,95,233,1,133,95
302 DATA 133,251,165,96,233,0,133,
    252,133,96,165,46,133,91,165,4
    5,133
304 DATA 90,24,105,7,133,88,133,45
    ,165,46,105,0,133,46,133,89,32
306 DATA 184,163,160,0,185,111,192
    ,145,251,200,192,7,208,246,32,
    51,165
308 DATA 24,165,251,105,8,170,165,
    252,105,0,133,252,138,133,251,
    160,0
310 DATA 177,251,133,95,200,177,25
    1,133,96,177,95,208,168,165,45
    ,133,47
312 DATA 133,49,165,46,133,48,133,
    50,96,58,143,34,141,147,34,204
```

Protezione Turbo Software

La protezione del proprio software dalla copia abusiva è un campo che prima o poi interessa tutti. Un semplice programma richiesto da più parti

di Giovanni Verrelli

Da qualche tempo la rivista "Commodore Club" su cassetta pubblica i propri programmi registrati col "Turbo Tape", ausilio indispensabile per gli utenti del Commodore 64 e registratore che permette di aumentare di circa 10 volte la velocità di caricamento da cassetta.

Quasi tutti possiedono tale programma e chi ne fosse ancora sprovvisto può rimediarvi acquistando una copia della rivista su cassetta prima citata.

Di seguito riportiamo un programma che permette la protezione del software registrato, appunto, in "turbo".

Per ovvi motivi saranno fornite solamente le istruzioni per un suo corretto uso in quanto spiegarne anche il funzionamento sarebbe controproducente ai nostri fini.

Cominciamo subito con l'elencare le varie fasi da seguire alla lettera per evitare ogni possibile problema:

1) caricare in memoria il "Turbo Tape", attivarlo col RUN e dare il comando NEW;

2) copiare il listato pubblicato di seguito, controllarlo fino ad avere l'assoluta certezza che non contenga errori e salvarlo (prima di provarlo) sulla memoria di massa posseduta. Lo si può tranquillamente registrare anche in turbo;

3) inserire nel registratore la cassetta che dovrà ospitare i due programmi generati dalla procedura, il primo in linguaggio macchina per effettuare il caricamento e l'avviamento automatico e l'altro che non sarà altro che il programma che si vuol proteggere, codificato;

4) dare il RUN al programma che si è appena copiato e, dopo una piccola schermata introduttiva, rispondere alle varie domande che si susseguiranno sullo schermo. Queste sono volte alla conoscenza del modo in cui parte il programma da proteggere (col RUN oppure con SYS) e del nome che gli vogliamo assegnare;

5) dopo l'ultima richiesta, comparirà il messaggio di premere i tasti RECORD & PLAY sul registratore. Farlo dopo essersi assicurati che la cassetta contenuta nel registratore sia effettivamente quella che dovrà ospitare il programma protetto. Al termine, togliere dal registratore la cassetta e metterla da parte;

6) seguire le istruzioni che compariranno sul video, e precisamente:

a) caricare il programma che si vuol proteggere;
b) digitare SYS 733 e premere Return;
c) reinserire la cassetta dove è stata effettuata la prima registrazione, battere il comando precisato sul video (oppure posizionarsi su di esso col cursore) e premere il tasto Return;
d) Premere i tasti RECORD & PLAY sul registratore;

A questo punto il nostro lavoro è terminato.

Per l'uso basterà semplicemente:

- caricare il programma utility "Turbo Tape" e attivarla.
- caricare con un normale LOAD (e non <-L) il primo programma registrato e premere la barra spaziatrice quando richiesto.

Come utilizzare la procedura

Facciamo ora un esempio chiarificatore.

Supponiamo di voler proteggere il programma "MOTOR MANIA", la cui attivazione è resa possibile dal comando SYS 8000 invece del normale RUN.

Per prima cosa caricate in memoria il "Turbo tape" e attivatelo (vale a dire: date il RUN e, non appena ritorna il cursore, NEW).

Fatto questo, predisponete la cassetta che ospiterà il programma protetto, caricate in memoria il programma per le protezioni (riportato in queste pagine) e attivatelo con un semplice RUN.

Alla prima domanda, con la quale si chiede se il programma da proteggere parte col RUN, digitate NO e premete il tasto Return.

Alla domanda successiva, dove è richiesta la SYS di partenza, rispondete con 8000 e Return.

Infine, alla richiesta del nome del programma da proteggere, rispondete battendo MOTOR MANIA.

Subito dopo la pressione del tasto Return, compare la richiesta di premere RECORD & PLAY sul registratore. Fatelo dopo aver inserito la cassetta preparata precedentemente.

Trascorsi circa 20 secondi di registrazione, seguite le istruzioni comparse sul video: Caricate in memoria il programma "MOTOR MANIA" originale, digitate SYS 733 (e Return) e, dopo aver reinserito la cassetta messa precedentemenet da parte, scrivete su video <-S, aprite i doppi apici (le virgolette), premete Shift+Clr/home, cursore giù, cursore a destra, scrivete PREMERE SPACE, chiudete i doppi apici ed inserite il ,1,1 finale.

Fatto tutto questo, premete il tasto Return, i tasti RECORD & PLAY sul registratore. La protezione è stata applicata al vostro programma.

Volendo utilizzare questo programma protetto, si carica in memoria il Turbo Tape, si riporta la cassetta all'inizio della prima memorizzazione, si effettua un LOAD "MOTOR MANIA" + Return, si preme il tasto Commodore al momento del FOUND e la barra spaziatrice quando compare il messaggio premere SPACE.

Tutto il resto, attivazione compresa, è automatico ed il compito finale è solo quello di utilizzare il programma dato che la partenza (RUN oppure SYS a seconda dei casi) è automatica.

Per quanto riguarda l'aspetto tecnico, tutta la procedura, oltre a decodificare ed avviare automaticamente il programma principale, si occupa di disabilitare il RUN/STOP, RESTORE, LIST, SAVE, <-S ed anche a controllare se, durante il caricamento del programma principale, sia intervenuto un ?LOAD ERROR, molto frequente col "Turbo". In questo caso si arresta e non dà l'avvio automatico del programma che potrebbe avere conseguenze non certo piacevoli.

In quest'ultimo caso l'unico rimedio è di riprovare a caricare il tutto e agire, al massimo, sulla vite di regolazione della testina del registratore.

Concludiamo avvertendo che, come sicuramente avrete già capito, questa procedura è valida per qualsiasi programma (in Basic o in linguaggio macchina) compatibile col Turbo Tape (ma non può proteggere se stesso). Qualche problema può insorgere solamente con pochissimi programmi che partono con la SYS 64738 di reset e che, anche per i programmi in linguaggio macchina che partono col RUN, è possibile effettuare le protezione basata sulla SYS.

Inconvenienti possibili

Con programmi molto lunghi o che occupano locazioni di memoria molto alte può capitare, mentre si cerca di salvare col comando <-S.... il programma codificato, di veder stampato sullo schermo il messaggio ?OUT OF MEMORY ERROR con il conseguente ritorno del cursore.

In questo caso una prima soluzione è quella di inserire, al posto del <-S precisato sullo schermo, il comando <-S "",1,1.

L'altra soluzione, che consigliamo in quanto non intacca minimamente lo spirito della procedura, è di leggere il puntatore all'inizio delle variabili con il comando:
PRINT PEEK(45)+PEEK(46)*256

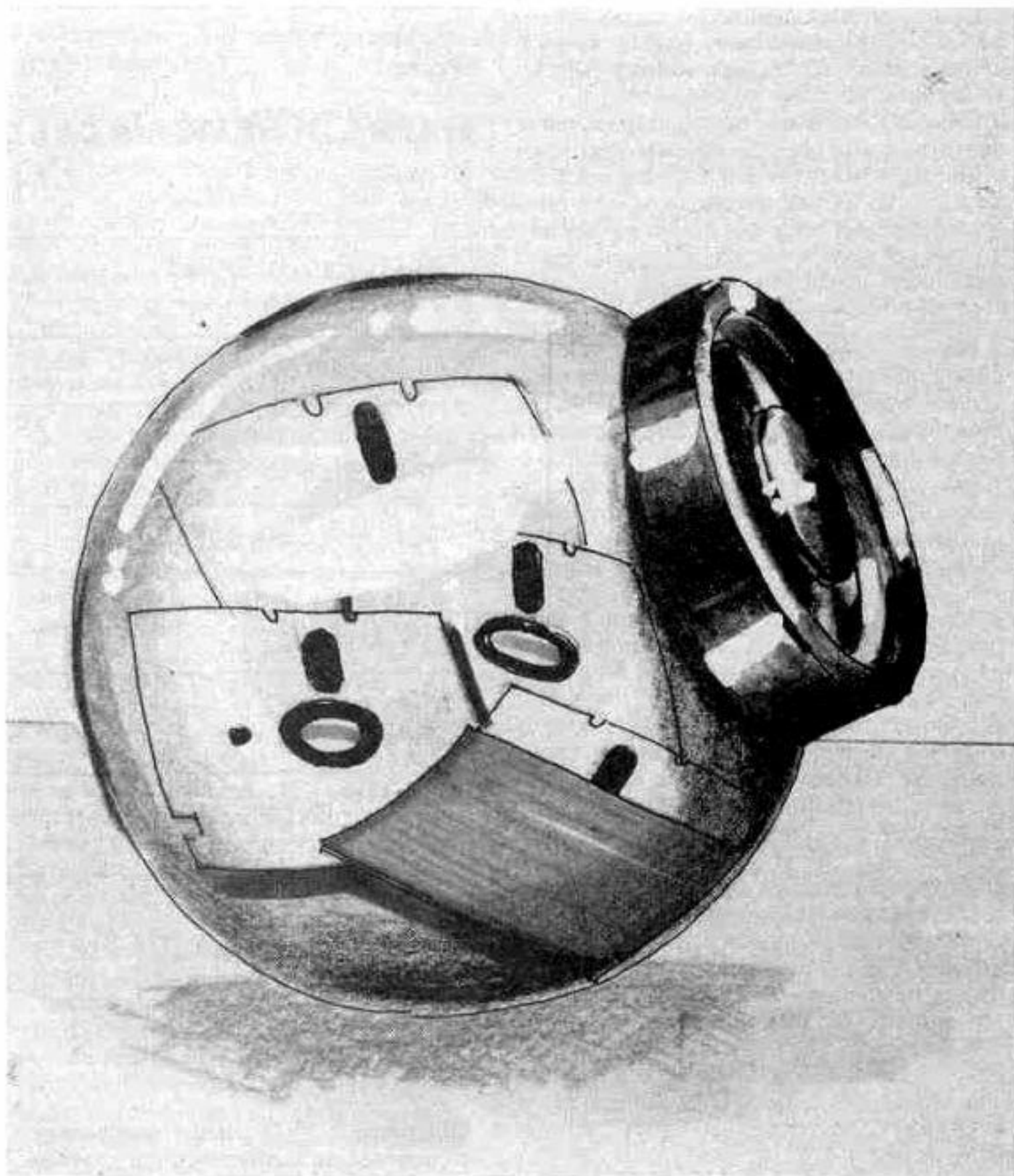
Se il valore che ne risulta è inferiore a 52084 (quasi sempre), allora basterà dare il comando POKE 56,200 e poi dare il <-S originario, ricordandosi alla fine di spegnere il tutto oppure di impartire la SYS 64738 di reset.

Nel caso in cui tale valore fosse superiore a 52084, allora si potrà solamente adottare la prima soluzione in quanto la seconda provocherebbe pericolose sovrapposizioni.

```
 10 REM   *************************
 20 REM   ** PROTEZIONE PROGRAMMI *
 *
 30 REM   ** REGISTRATI IN  TURBO *
 *
 40 REM   **   PER COMMODORE 64   *
 *
 50 REM   ** BY GIOVANNI VERRELLI *
 *
 60 REM   ** GIOIA DEL COLLE (BA) *
 *
 70 REM   *************************
 *
 75 REM  NON RENUMERARE IL PROGRAM
 MA
 80 POKE 53280,0:POKE 53281,0
 90 PRINTCHR$(159);CHR$(147)
100 PRINT"[RIGHT][RVS]      PROTEZI
 ONE      [RVOFF]"
110 PRINT"[RIGHT][RVS]   TURBO SOF
 TWARE   [RVOFF]"
120 PRINT"[RIGHT][RVS]BY GIOVANNI
 VERRELLI[RVOFF]"
130 PRINT"[RIGHT][RVS]GIOIA DEL CO
 LLE (BA)[RVOFF]"
140 CK=0:FOR I=679 TO 767:READ GV:
 POKE I,GV:CK=CK+GV:NEXT
150 IF CK<>10241 THEN PRINT"ERRORE
  NEI DATA: CONTROLLARE.":END
160 A$="":PRINT"IL PROGRAMMA DA PR
 OTEGGERE"
170 INPUT "PARTE COL [RVS]RUN[RVOF
 F] (S/N)";A$
180 IF LEFT$(A$,1)="S" THEN 270
190 SY=0:PRINT"QUAL E' LA [RVS]SYS
 [RVOFF] DI PARTENZA";:INPUT SY
200 IF SY<1 OR SY>65536 THEN GOSUB
  500:GOTO 190
210 F=0:IF SY=64738 THEN SY=32768:
 F=1
220 HI=INT(SY/256):LO=SY-HI*256
230 POKE 736,LO:POKE 754,LO
240 POKE 737,HI:POKE 755,HI
250 IF  F  THEN LO=226:HI=252
260 POKE 759,76:POKE 760,LO:POKE 7
 61,HI
270 A$="":PRINT"QUAL E' IL NOME DE
 L PROGRAMMA"
280 INPUT "DA PROTEGGERE";A$
290 IF LEN(A$)>16 THEN GOSUB 500:G
 OTO 270
300 POKE 646,PEEK(53281)
310 PRINTCHR$(147)"400 SAVE"CHR$(3
 4);A$;CHR$(34);",1,1"
320 PRINT"RUN350"
330 FOR P=1 TO 10:GET P$:NEXTP
340 POKE 631,19:POKE 632,13:POKE 6
 33,13:POKE 198,3:END
350 PRINTCHR$(147);CHR$(159)
360 POKE 770,167:POKE 771,2
370 POKE 1020,PEEK(43):POKE 1021,P
 EEK(44)
380 POKE 1022,PEEK(45):POKE 1023,P
 EEK(46)
390 POKE 43,167:POKE 44,2:POKE 45,
 4:POKE 46,3
400 REM  ** COPIA ANCHE QUESTA LIN
 EA **
410 POKE 770,131:POKE 771,164
420 POKE 43,PEEK(1020):POKE 44,PEE
 K(1021)
430 POKE 45,PEEK(1022):POKE 46,PEE
 K(1023)
440 POKE 759,96
450 PRINTCHR$(147)"PER COMPLETARE
 LA PROCEDURA CARICA IL"
460 PRINT"PROGRAMMA CHE VUOI PROTE
 GGERE, BATTI IN"
470 PRINT"DIRETTO [RVS]SYS 733[RVO
 FF] E POI EFFETTUA UN"
480 PRINT"←S"CHR$(34)CHR$(147)CHR$
 (17)CHR$(29)"PREMERE SPACE"CHR
 $(34)",1,1"
490 NEW
500 PRINT"QUALCOSA NON VA: RIPETI.
 ":RETURN
510 DATA 169,225,141,040,003,169,0
 20,141,109,195
520 DATA 169,244,141,051,003,169,1
 95,141,002,003
530 DATA 169,002,141,003,003,076,1
 33,195,169,131
540 DATA 141,002,003,169,164,141,0
 03,003,165,144
550 DATA 240,003,108,002,003,165,1
 74,133,045,165
560 DATA 175,133,046,234,162,000,1
 89,000,008,208
570 DATA 009,232,240,016,232,240,0
 13,076,244,002
580 DATA 073,255,240,003,157,000,0
 08,232,208,232
590 DATA 032,094,166,032,142,166,0
 76,174,167
```

Autostart 64

di Pasquale D'Andreti

Una semplice e breve utility per proteggere i propri programmi, che potrà essere utilizzata, limitandosi a digitarlo e a seguire le semplici istruzioni, anche da coloro che non intendono "studiare" il Linguaggio Macchina.

A parecchi sarà capitato di caricare un programma da nastro o da disco e di constatare come questo, appena terminata la fase di lettura, parta da solo, come se il computer avesse digitato, da solo, RUN + Return. Naturalmente, oltre ad essere comoda per la brevità, questa anormale procedura di caricamento impedisce all'utente di "sbirciare" nel programma e di apportarvi eventuali modifiche. Spesso un programma registrato con la tecnica dell'AUTOSTART (partenza automatica) non può essere bloccato con la pressione del tasto RUN/STOP, nemmeno in compagnia del più potente RESTORE. Sarebbe molto interessante applicare questa tecnica ai nostri programmi, ma soprattutto sapere come si comporta il 64 quando carica questo tipo di file.

Per fare ciò, parleremo prima di diversi argomenti che, apparentemente non connessi da un filo logico, saranno di fondamentale importanza per il raggiungimento del nostro fine. Solo in seguito potremo descrivere il funzionamento della routine che genera l'AUTOSTART.

Una panoramica sul basic del 64

Appena acceso, il C-64 si presenta con la nota videata avvertendoci, con il READY, che è pronto ed aspetta che noi immettiamo, tramite la tastiera, un comando o una linea scritti in basic, convalidata dalla pressione del tasto Return.

Ma che cos'è e come funziona il basic del 64?

Il microprocessore 6510, cioè la "mente" del computer, non capisce il basic, ma solo il Linguaggio Macchina (L.M.), linguaggio non facile da apprendere e da usare, ma velocissimo ed efficiente. Pertanto, prima di eseguire un qualsiasi comando basic, il 6510 ha bisogno di un programma in Linguaggio Macchina che "traduca" il comando in una serie d'istruzioni direttamente eseguibili dal microprocessore.

In altre parole, il 6510 ha bisogno di un "interprete".

Questo traduttore è appunto costituito da un programma in L.M. lungo 8 Kbytes che, grazie alle ROM possedute dal 64 (memorie che non si cancellano), risiede permanentemente nel computer, attivandosi ogni volta che lo accendiamo. Il basic fa uso di alcune routine di utilità generale (tra cui quelle che gestiscono la registrazione e la lettura dei programmi), che fanno parte del KERNAL (nocciolo), anch'esso residente in ROM, per altri 8 Kbytes di memoria.

I vettori

La mappa della memoria del C-64, quando questo viene acceso, si presenta come in figura 1. Come si può notare, essa comprende 65536 locazioni di memoria le quali sono divise in tre gruppi principali:

1/ memoria RAM: si può leggere e ci si può scrivere, ma il suo contenuto viene perso quando si spegne il computer (memoria detta, appunto per questo, "volatile");
2/ memoria ROM: si può solo leggere, ma è permanente;
3/ memoria I/O: controlla il funzionamento delle periferiche, del video e del suono.

Come precedentemente affermato, il basic ed il KERNAL risiedono su ROM e, essendo immodificabili, dobbiamo sorbirci passivamente tutto ciò che ci rifilano.

Ed infatti è così se si lavora in basic, ma se siamo in grado di scrivere delle routine in L.M. che si sostituiscano a quelle del basic fornito dalle ROM, possiamo mutare il comportamento del computer a nostro piacimento. Ma, visto che le routine del basic non sono modificabili, in che modo possiamo realizzare lo scopo prefissato?

Mamma Commodore ha pensato anche a questo e ha creato tabelle di vettori (detti anche "puntatori") che risiedono in RAM. Spieghiamoci meglio: il L.M. che costituisce il basic è dotato di diverse decine di routine a se stanti che realizzano determinate funzioni (come la stampa sul video di un certo messaggio, la decifrazione dei comandi, la loro esecuzione, il controllo degli errori eccetera) e le più importanti sono state "vettorizzate" in RAM. Ad esempio, quando il programma principale che controlla tutte queste routine ha "scoperto" che in un comando da noi digitato c'è un errore, non chiama direttamente la routine che li gestisce, bensì legge il contenuto di due locazioni di due locazioni di memoria RAM consecutive (in questo caso la 768 e la 769), saltando alla routine che inizia a partire dall'indirizzo contenuto in questi due byte.

Precisiamo che ci vogliono due byte per codificare un indirizzo, perchè un byte può contenere valori che vanno da 0 a 255 mentre sappiamo che un indirizzo di memoria può arrivare fino a 65535 (ulteriori chiarimenti su questo argomento si trovano nell'articolo "Microgioco da addestramento", comparso sul numero 23 - Settembre 1985 - di questa rivista). Il valore contenuto nel vettore si calcola leggendo il secondo byte, moltiplicandolo per 256 ed aggiungendo il valore del primo byte.

Supponiamo, per esempio, che la locazione 768 contiene il valore 139 e la 769 il valore 227. Se si moltiplica 227 per 256 (ottenendo 58112) e si aggiunge 139, si ottiene l'indirizzo desiderato: 58251. In pratica ciò si ottiene digitando:

```
PRINT PEEK(769)*256 + PEEK(768)
```

Se, invece di leggerlo, vogliamo depositare un certo valore in un vettore, dovremo dividerlo per 256, porre il quoziente nel secondo byte ed il resto nel primo. Esempio: se vogliamo porre l'indirizzo 2049 nel vettore 700-701, dovremo dividere 2049 per 256, ottenendo 8 come quoziente intero ed 1 come resto; 8 va in 701 ed 1 in 700.

Una schematica rappresentazione di quello che è stato descritto è offerta dalla figura 2. Ricordiamo che tutto ciò avviene sempre ed automaticamente tutte le volte che il basic deve eseguire una routine vettorizzata e non solo quella che gestisce gli errori, che abbiamo riportato a puro titolo di esempio.

Adesso è possibile capire come si possa intervenire direttamente sul funzionamento del basic: sarà sufficiente modificare i puntatori di queste routine in modo che indirizzino ad una locazione RAM e scrivere, a partire da questa, una nostra routine in L.M.

Figura 1

MAPPA DI MEMORIA DEL C-64

RAM usata dal BASIC e dal KERNAL	0 1023
RAM usata dal video	1024
RAM utilizzabile per i programmi BASIC	2048 40959
ROM interprete BASIC	49151
RAM libera a disposizione dell'utente	49152 53247
Memoria di I/0 i RAM colore	53248 57343
ROM KERNAL	57344 65535

Il buffer della tastiera

Digitate questo semplice programmino:

```
100 IF PEEK(198) < > 10 THEN 100
```

Dando il RUN, il cursore scompare perchè il computer sta eseguendo un programma. Le istruzioni della linea 100 significano "aspetta finchè il valore contenuto nella locazione 198 non sia 10". A questo punto digitiamo lentamente dieci caratteri a caso notando che sul video, finchè non abbiamo premuto il decimo tasto, non accade nulla. Poi appare il READY segnalando che il programma è terminato e, quasi contemporaneamente, i caratteri che abbiamo digitato precedentemente vengono visualizzati, come se il computer li avesse "ricordati". La stessa cosa succede se il programma è scritto in L.M. Il C-64 si "ricorda" dei caratteri digitati perchè, a mano a mano che vengono scritti dall'utente, esso li deposita in una parte di memoria chiamata BUFFER, situata in RAM dalla locazione 631 alla 640.

Il numero massimo di caratteri che si possono immagazzinare in questo buffer è 10 e questo valore, contenuto nella locazione 198, viene automaticamente aggiornato dal computer a mano a mano che si premono i tasti. Ora si comprende il funzionamento del programmino: esso aspetta che il buffer si sia riempito (cioè che la locazione 198 contenga 10) prima di "uscire". Naturalmente, essendo in RAM, il buffer è modificabile dall'utente che deve seguire questa procedura:

1/ Depositare, mediante POKE, i codici dei caratteri prescelti nel buffer (il primo va in 631, il secondo in 632 e così via, fino alla 640). Precisiamo che i codici sono del tipo ASCII e la corrispondente tabella carattere-codice si trova nell'Appendice F del manuale d'istruzioni;

2/ introdurre il numero dei caratteri memorizzati nel buffer nella locazione 198 (per esempio, nel caso si fossero utilizzati 7 codici, ci vuole POKE 198,7);

3/ fare in modo che compaia READY sullo schermo (per esempio, far terminare il proprio programma con END o STOP).

Con questo programmino, comparirà la scritta "64" sullo schermo:

```
10 PRINT CHR$(147): "SONO IL COMMODORE"
20 POKE 631,54: REM 54 è il codice ASCII di "6"
30 POKE 632,52: REM 52 è il codice ASCII di "4"
40 POKE 198,2: REM 2 è il numero dei caratteri di "64"
50 END
```

Struttura e collocazione dei programmi basic

Ogni volta che immettiamo una linea basic, il computer la memorizza in una zona di memoria che inizia all'indirizzo 2049 e può terminare all'indirizzo 40959. La prima linea è quindi posta a partire da 2049. Quando digitiamo LIST, il computer esamina questa zona

di memoria stampando sul video ciò che incontra, finchè incontra tre zeri consecutivi che indicano appunto la fine del programma. Nel caso che le locazioni 2049, 2050 e 2051 contengano proprio tre zeri, vuol dire che non c'è programma in memoria e pertanto appare solo READY (Figura 3). E' ovvio che i tre zeri di seguito li trascrive il computer tutte le volte che è necessario (aggiunta, modifica, cancellazione di linee basic).

Qualcosa di più sul LOAD e sul SAVE

In genere, quando leggiamo un programma da nastro (LOAD "NOME") o da disco (LOAD "NOME",8), esso viene caricato automaticamente a partire da 2049 (caso del C-64), presupponendo che sia un programma in basic anche se in effetti era stato registrato da una zona di memoria diversa da quella adibita ai programmi basic.

Se noi digitiamo LOAD "NOME",1,1 o LOAD "NOME",8,1 il programma si caricherà esattamente nella zona di memoria da dove era stato prelevato.

C'è anche un caso, utilizzato dalla routine AUTOSTART, per cui, anche digitando LOAD "nome" senza porre il ",1,1" finale, il programma si caricherà a partire dalla stessa locazione da dove era stato registrato. Purtroppo ciò non vale per il drive, che ha bisogno del ",8,1" per caricare un programma senza rilocarlo.

Per salvare programmi in basic, basta il SAVE "NOME" (con il solito ",8" finale per il drive), che salva il nostro programma incominciando da 2049.

Ci sono diverse tecniche per registrare porzioni di memoria che iniziano in locazioni diverse dalla 2049, ma la descrizione di queste esulano dallo scopo dell'articolo. Ci basti sapere che il programma AUTOSTART usufruirà proprio di una di tali tecniche per registrare una piccola parte di memoria diversa da quella abitualmente dedicata ai programmi basic.

Descrizione e funzionamento del programma

Tra le routine del basic vettorizzate figura la WARM START ROUTINE, che è quella che il sistema esegue quando stampa READY (per brevità ROUTINE READY). Pertanto, quando il basic del 64 visualizza READY, vuol dire che è stata utilizzata questa routine, la quale ha il suo vettore in 770 e 771. Il valore di questo puntatore, calcolato con il solito metodo:
PRINT PEEK(771)*256 + PEEK(770)
è uguale a 42115, numero corrispondente alla locazione d'inizio della routine READY.

Modificheremo questo puntatore in modo che indirizzi ad una routine in L.M. scritta da noi. Questa, ad esempio, avrà il compito di "forzare" i codici di "64" nel buffer di tastiera, ottenendo, come risultato, che ogni volta che il computer eseguirà la routine READY, comparirà il numero "64".

Possiamo notare che il READY appare anche dopo il caricamento di un programma; pertanto per realizzare il nostro AUTOSTART dovremo scrivere una routine in L.M. che forzi nel buffer di tastiera il comando RUN+Return dopo aver letto il programma in basic e modificare il vettore di READY in modo che punti alla routine sopracitata. Scriveremo questa routine in una zona di memoria libera che va dalla locazione 679 alla 767, proprio al di sotto della tabella dei vettori (figura 4).

Si tratta, dunque, di registrare due programmi in successione sullo stesso nastro.

Tutto ciò potrebbe essere facilmente raggirato da chi desideri copiare e/o modificare il nostro programma: basterebbe infatti leggere solo il secondo programma. Per impedire che ciò avvenga, dovremo utilizzare due tecniche di protezione, la prima delle quali disabilita il tasto RUN/STOP e la funzione LIST (che si ottiene ponendo il valore 227 nella locazione 808) mentre la seconda mette a zero i bytes che vanno da 2049 a 2059 facendo in modo che, anche leggendo solo il secondo programma, non sia nè listabile nè eseguibile. Ricordiamo, infatti, che quando i bytes 2049,2050 e 2051 sono posti a zero, il basic "crede" che non vi sia alcun programma in memoria.

Riorganizziamo le idee e descriviamo, passo passo, il funzionamento della routine AUTOSTART una volta caricato in memoria il programma basic che si intende proteggere:

1/ Riporta nella zona di memoria che va da 679 a 689 i bytes che vanno da 2049 a 2059 (cioè i primi dieci byte della parte iniziale del programma basic) azzerando questi ultimi;
2/ trasferisce la routine LOADER nella memoria che va da 690 a 767 (il funzionamento della stessa verrà spiegato più avanti);
3/ modifica il vettore di READY facendo in modo che punti alla routine LOADER (l'indirizzo di LOADER è 690, pertanto pone 2 in 771 e 178 in 770);
4/ salva la zona di memoria compresa tra 679 e 771; nel caso fosse utilizzato il nastro, la registrazione sarà effettuata in maniera tale che, anche omettendo il ",1,1" dopo il LOAD, la routine si posizioni automaticamente nella zona da cui era stata prelevata.

Compiute queste operazioni, non ci resta che registrare con un normale SAVE il nostro programma protetto.

Vediamo ora che cosa succede quando si carica un programma protetto con la tecnica dell'AUTOSTART.

Dando il LOAD, il 64 carica il programma che incontra per primo sul nastro, vale a dire il LOADER, nella zona di memoria che si estende dalla locazione 679 alla 771. Terminata la lettura, il computer "vorrebbe" eseguire la routine READY cercandone l'inizio nel vettore 770-771. Questo però, dopo il caricamento, non vale più 42115, ma 690, cioè proprio l'indirizzo della nostra routine LOADER. Una volta attivata, la routine per prima cosa legge il programma protetto, poi rimette "a posto" i bytes 2049-2059 prelevando i valori originali dalle locazioni 679-689; così facendo rende il programma basic perfettamente utilizzabile.

Figura 2

IL VETTORE

Locazione	Contenuto		Indirizzo
769	227	x 256 =	58112
768	184	x 1 =	184
		TOTALE =	58296

OPERAZIONE INVERSA

58296:256 = 227.71875 — **da cui 227**
.71875*256 = 184 — **da cui 184**

Figura 3

IL PROGRAMMA BASIC

Contenuto	Indirizzo
0	2048
???	2049
???	2050
------	--------
???	Fine programma
0	" " + 1
0	" " + 2
0	" " + 3

Figura 4

LA ROUTINE LOADER

Dati BASIC	679 689
Routine in L.M.	690 767
178 2	770 771
184	768
227	769
Vettore READY (770/771) 2x256+178=690	

Dopo aver eseguito queste operazioni, rimette il valore primitivo (42115) nel vettore di READY, pone 227 nella locazione 808, disabilitando il tasto Run/stop, e forza nel buffer di tastiera il comando RUN+Return saltando, infine, alla originale routine di READY.

Dopo l'esecuzione di tale routine, appare il famoso READY, ma, come spiegato precedentemente, il computer preleva gli eventuali caratteri presenti nel buffer di tastiera i quali sono appunto "RUN"+Return. In questo modo il programma basic si avvia automaticamente e non può essere più fermato, essendo inibito il tasto RUN/STOP.

Tutte le operazioni compiute dalla routine LOADER sono eseguite con una tale fulmineità da impedire all'operatore di accorgersi di ciò che sta succedendo.

Finalmente, dopo le doverose premesse teoriche, passiamo ai fatti.

Come proteggere un programma

Per prima cosa bisogna digitare il listato pubblicato in queste pagine, verificarne la corretta trascrizione e registrarlo, col nome di "AUTOSTART", su una cassetta o su un dischetto a parte, preferibilmente all'inizio, in modo che sia di facile reperibilità. Per velocizzare la copiatura del listato, si possono tralasciare tutte le linee che iniziano con l'istruzione REM o con i due punti. Nella trascrizione, bisogna porre particolare attenzione alle linee DATA; in ogni caso il programma si accorge da solo della presenza di eventuali errori in queste linee visualizzando l'apposito messaggio. Per i lettori più esperti, il listato, opportunamente disassemblato, servirà come base per uno studio più approfondito della tecnica dell'AUTOSTART e per facilitare eventuali personalizzazioni.

Benchè superfluo, ricordiamo ai principianti (che intendono utilizzare la tecnica di protezione senza studiarne il funzionamento) che devono limitarsi a digitare il programma basic e seguire le istruzioni fornite qui di seguito.

Prima di descrivere le operazioni da fare per dotare di AUTOSTART un qualsiasi programma scritto in basic, è bene fare alcune premesse:

- Il programma basic pubblicato in queste pagine lo chiameremo "AUTOSTART".
- Il programma che intendete proteggere deve essere registrato secondo le procedure normalissime (sul disco oppure nastro N.1) col nome "PROVA" o con altro nome di fantasia.
- Il supporto magnetico (disco o nastro) destinato ad ospitare il programma protetto (somma, in realtà, di due programmi) lo chiameremo Disco (Nastro) N.2.

Passiamo, ora, alla descrizione delle operazioni da compiere, e che sono le seguenti:

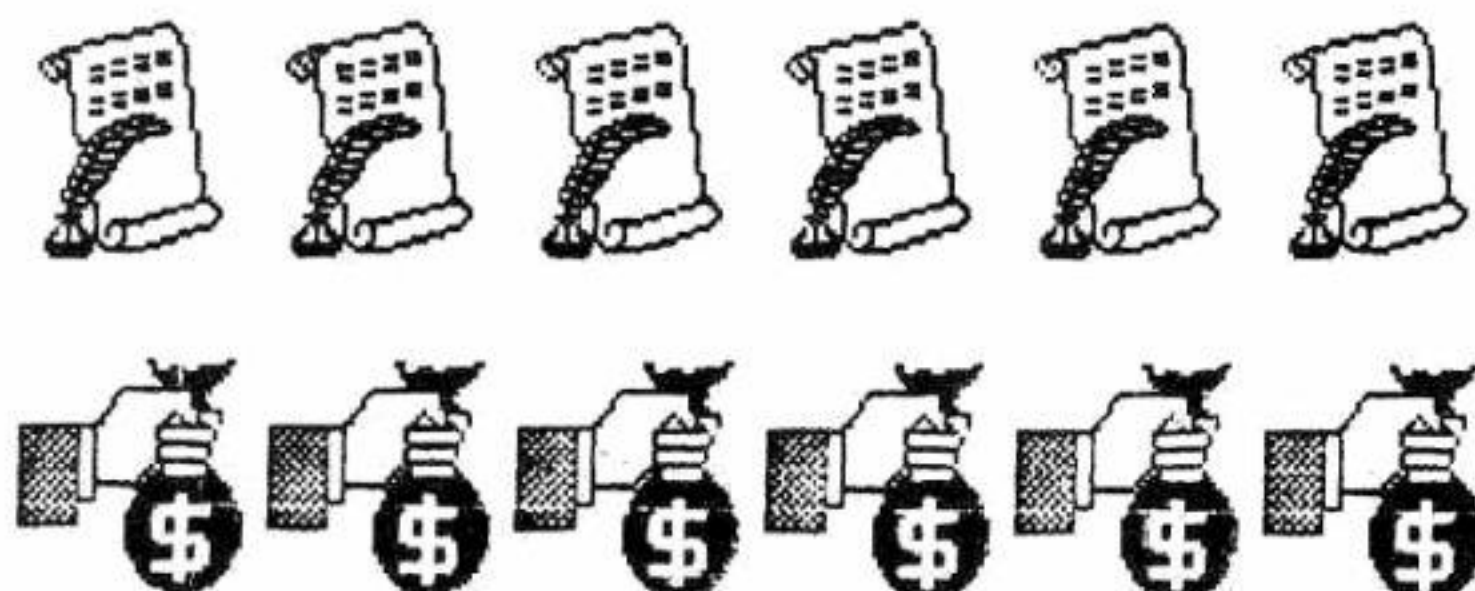
1/ caricare (da disco o nastro) la routine AUTOSTART;
2/ dare il RUN e, alla domanda "NASTRO O DISCO?", rispondere con "N" oppure "D" a seconda del supporto su cui desideriamo registrare il programma protetto, non dimenticando di premere il tasto Return;
3/ caricare (dalla unità nastro o disco selezionata) il programma "PROVA" che si vuole proteggere. Chi utilizza il drive dovrà scrivere il nome per intero (se, ad esempio, il programma da proteggere si chiama "PROVA", si deve leggere con LOAD "PROVA",8 e NON con LOAD "PRO*",8);
4/ inserire il nastro (o il dischetto) N.2, e digitare SYS 49152+Return per provocare la registrazione del LOADER;
5/ attendere la fine della registrazione e, se si usa il registratore, riavvolgere il nastro di 3 giri (porre la massima attenzione nel compiere questa operazione). In qualunque caso non cambiare il supporto (che è ancora il N.2);
6/ quando compare il READY che indica il termine della registrazione del LOADER, digitare, senza cambiare il supporto magnetico, il comando SAVE "1ROVA" (+",8" se si usa il drive) e premere il tasto Return. Il nome del programma da proteggere dovrà essere uguale al titolo originale tranne il primo carattere che deve essere il carattere "1" (ecco perchè il nome "PROVA" diventa "1ROVA". Pertanto bisognerà digitare SAVE "1ROVA" oppure SAVE "1ROVA",8 nel caso del disco).

Dopo aver effettuato la registrazione descritta al punto 6, il gioco è fatto e per verificare la correttezza della procedura, riavvolgere il nastro (al disco non serve!) e caricare il programma con LOAD (registratore) oppure LOAD "PROVA",8,1 (disco).

Ora non dobbiamo far altro che attendere: il primo programma, partendo automaticamente, caricherà il secondo e, resolo eseguibile, gli darà l'"irreversibile" RUN+Return. Se si usa il drive e si omette il ",1" finale, chiedendo il listato si otterrà una fila di caratteri insignificanti. Agli utilizzatori del registratore, giustamente preoccupati dal fatto che riavvolgendo il nastro di tre giri si possa cancellare il LOADER, diciamo che quest'ultimo è salvato in appena sette giri, ma anzichè smettere di registrare, il 64 continua per un po' ad inviare al nastro caratteri privi di senso che potrebbero compromettere il buon fine della procedura e che pertanto devono essere eliminati.

Considerazioni

L'articolo contiene vari spunti di esercizio per il principiante (buffer di tastiera) e per l'esperto (gestione diretta della registrazione e della lettura dei programmi in L.M.), nonchè una più generale inquadratura schematica del funzionamento del 64.

I primi potrebbero, per esempio, cercare il modo di scrivere un programma in basic che, terminando, carichi un altro programma utilizzando appunto la tecnica del buffer di tastiera; i secondi potrebbero individuare un'altra tecnica di AUTOSTART (si pensi allo stack del 6510 e al modo di funzionare delle istruzioni RTS e RTI ...). Il materiale non manca, il computer nemmeno, bisogna solo dotarsi di buona volontà e iniziare a provare e riprovare.

```
  3 REM  PROTEZIONE CON
  4 REM  SISTEMA DI AUTOSTART.
  5 :
  6 REM  DI PASQUALE D'ANDRETI
  7 REM  RIARDO (CE)
  8 :
  9 Q$=CHR$(34):PRINTCHR$(147)"ATT
    ENDERE"
 11 REM "*** LETTURA DATA E MEMORI
    ZZAZIONE L.M.
 15 FOR  I = 49152  TO  49316: REA
    D  A: B=B+A: POKE  I,A: NEXT
 20 IF  B = 19763  THEN  50
 30 PRINT"ERRORE NEI DATA:"
 40 PRINT "RIVEDERE IL LISTATO.":
    END
 50 PRINT"NASTRO O DISCO (N/D)"
 55 GET ME$:IF ME$="" THEN 55
 60 IF  ME$<>"N"  AND  ME$<>"D"  T
    HEN  55
 70 IF  ME$="N"  THEN  ME$="NASTRO
    ":GOTO 90
 72 ME$="DISCO"
 77 REM  *** MODIFICA NUMERO PERIF
    ERICA E INDIRIZZO
 78 REM  *** SECONDARIO SE SI USA
    IL DRIVE
 80 POKE  49223,8: POKE  49225,1:
    POKE  49242,8
 87 :
 88 REM  *** ISTRUZIONI
 89 :
 90 PRINT CHR$(147)"ORA CARICA IL
    TUO PROGRAMMA, PREPARA  IL "ME
    $;
100 PRINT " (DIFFERENTE DA QUELLO
    ";
110 PRINT "ORIGINALE) , DIGITA  SY
    S 49152 + RETURN";
120 PRINT " E ASPETTA  CHE LA  ROU
    TINE VI REGISTRI";
130 PRINT " L'AUTOSTART. QUINDI DE
    VI";
135 IF ME$="DISCO" THEN 150
140 PRINT " DEVI RIAVVOLGERE IL NA
    STRO DI 3 GIRI E ";
150 PRINT "DARE IL SAVE IN MODO CH
    E IL NOME SIA UGUALE A  QUELLO
    ";
160 PRINT "USATO PER LEGGERLO TRAN
    NE  IL  PRIMO CA";
170 PRINT "RATTERE, CHE DEVE ESSER
    E UN "CHR$(34)"1"CHR$(34)"."
180 PRINT"AD ESEMPIO, SE HAI CARIC
    ATO IL PROGRAMMA";
190 PRINT "ORIGINALE CON  LOAD "Q$
    "TEST"Q$",  ORA  DOVRAI";
192 PRINT "SALVARLO CON SAVE "Q$"1
    EST"Q$"."
193 PRINT:PRINT"LOAD"Q$"...... -VE
    CCHIO "ME$
194 PRINT:PRINT"SYS 49152   -NUOVO
     "ME$:PRINT:PRINT"SAVE"Q$"1...
    .  -NUOVO "ME$
196 REM  ***
197 REM "*** CODICI COSTITUENTI IL
     L.M.
198 REM  *** INIZIO = 49152: FINE=
     49316
199 REM  ***
200 DATA 162,10,189,1,8,157,167,2,
    169,0,157,1,8,202,16,242
210 DATA 162,77,189,87,192,157,178
    ,2,202,16,247,169,178,141,2,3
220 DATA 169,2,141,3,3,32,68,192,1
    69,2,133,254,169,167,133,253
230 DATA 169,253,162,4,160,3,32,21
    6,255,169,164,141,3,3,169,131
240 DATA 141,2,3,96,169,1,162,1,16
    0,3,32,186,255,165,183,166
250 DATA 187,164,188,32,189,255,96
    ,169,1,162,1,168,32,186,255,16
    0
260 DATA 0,169,49,145,187,165,183,
    166,187,164,188,32,189,255,169
    ,0
270 DATA 32,213,255,134,45,132,46,
    162,10,189,167,2,157,1,8,202
280 DATA 16,247,162,4,134,198,189,
    251,2,157,118,2,202,208,247,16
    9
290 DATA 131,141,2,3,169,164,141,3
    ,3,169,227,141,40,3,76,131
300 DATA 164,147,82,117,13
```

...A PROPOSITO DI STAMPANTE, SEI SICURO D'AVER DATO

CARATTERISTICHE TECNICHE

- Velocità di stampa di 120 caratteri/secondo, bidirezionale e ottimizzata, monodirezionale in near letter quality e grafica.
- Testina di stampa a 9 aghi, larghezza carta 80 colonne.
- Matrice di stampa: 8x11 (standard), 8x6 (grafica a blocchi), 7x60 punti per pollice (grafica bit image), 17x11 (carattere definibile).
- Stampa in normale (10 CPI), condensato (17 CPI), NLQ (10 CPI), espanso (2, 3 o 4 volte), enfatizzato e italico.
- Alimentazione carta a moduli continui (trattore) e fogli singoli (frizione).
- Altre caratteristiche: segnalazione d'errore, caratteri definibili dall'utente, autotest, near letter quality, spoiler per strappo carta, grafica e indirizzamento di singoli punti, dump esadecimale.
- Compatibile con tutti gli Home Commodore: VIC20, C16, C64, C128, Plus 4.

adsGlen Milano

Probabilmente Star SG10C è la stampante ideale per l'utente avanzato di Commodore

che come te, vuole scrivere lettere, stampare tabulati, tracciare disegni...

il tutto velocemente e affidabilmente.

Totalmente compatibile con il Commodore C64, alfanumerica e grafica, con i suoi 120 caratteri/secondo SG 10C è un punto d'arrivo per i possessori di Commodore C64. Grazie alla possibilità di stampare anche caratteri near letter quality (selezione possibile con la semplice pressione di un tasto), SG 10C può essere asservita al Tuo Commodore come una macchina da scrivere.

Inviaci questo coupon, ti segnaleremo il nome e l'indirizzo di un rivenditore della tua zona che ti potrà mostrare la stampante Star SG10 C per il tuo Commodore, ma.........
ATTENZIONE!
Se deciderai di acquistare Star SG10 C l'invio del coupon ti darà il diritto ad uno SCONTO DEL 10%

LA TUA STAMPANTE

DISTRIBUTORE PER L'ITALIA

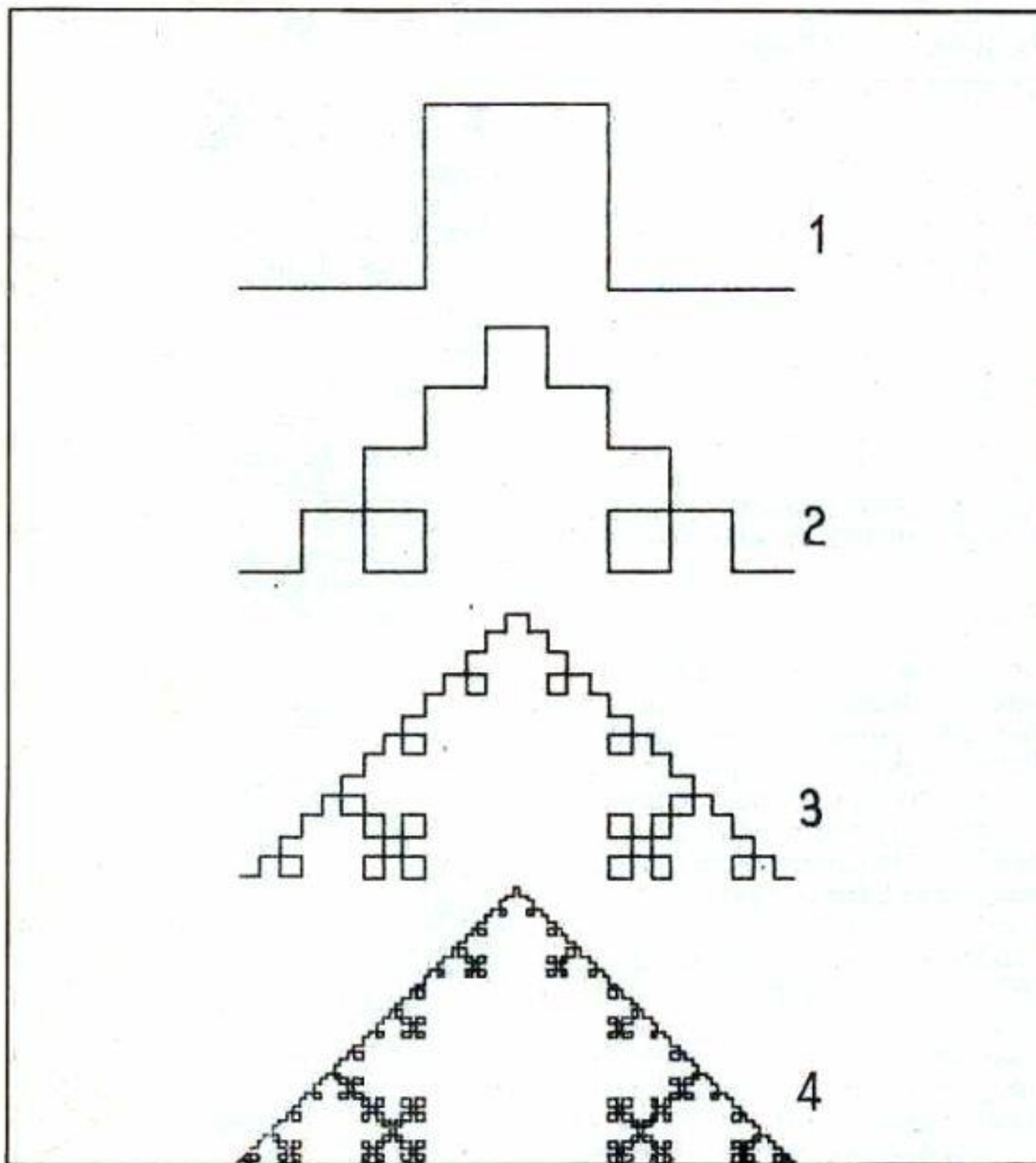
CLAITRON SpA

socio assodel

SEDE e UFF. COMM.: via gallarate 211 - 20151 milano
tel. (02) 301.00.81 - 301.00.91 (8 linee ric. aut.)
Telex n. 313843 CLAIMI

Ricorsività e merletti in Basic

di Danilo Toma

Una diabolica microprocedura BASIC per creare velocemente disegni stupefacenti! "Provare per credere..."

Creazione di una curva di Peano

Nella figura sono evidenziati i primi quattro ordini della procedura che, applicata all'infinito, fa tendere la figura ad una curva di Peano.

Come procurarsi le routine grafiche

I nuovi lettori di Commodore Computer Club forse non sono a conoscenza del fatto che la nostra rivista ha pubblicato tempo fa (esattamenteisul N.14) un programma in linguaggio macchina che aggiunge nuove, potenti istruzioni BASIC al C-64.

Queste routine, esclusivamente grafiche, consentono di disegnare in tre dimensioni in alta risoluzione.

Digitare il programma è piuttosto lungo e pericoloso e pertanto sconsigliamo, ai lettori inesperti, la sua digitazione.

Per venire incontro alle loro esigenze, tuttavia, la Systems Editoriale invia su richiesta, in cassetta, le routine grafiche insieme a tutti i programmi pubblicati sul N.14 di C.C.C.

Per avere la cassetta è sufficente inviare la modica cifra di 5800 (cinquemila ottocento) lire indicando, ovviamente, sia il numero di C.C.C. desiderato sia il vostro nominativo completo. Non si effettuano spedizioni contrassegno.

SYSTEMS EDITORIALE
Servizio arretrati
Viale Famagosta 75
20142 MILANO

Avete mai sentito parlara delle curva di Peano? La maggior parte di voi sicuramente no, a meno che non siate degli appassionati di matematica.

Neppure io le conoscevo, fino a quando lessi un articolo dedicato a tale argomento.

La curiosità mi spinse a realizzare un programma che potesse disegnare alcune di queste curve, per due motivi: volevo constatare personalmente la spettacolarità di tali "mostruosità" matematiche e, soprattutto, mi stuzzicava la dichiarata impossibilità di implementare tali procedure in BASIC.

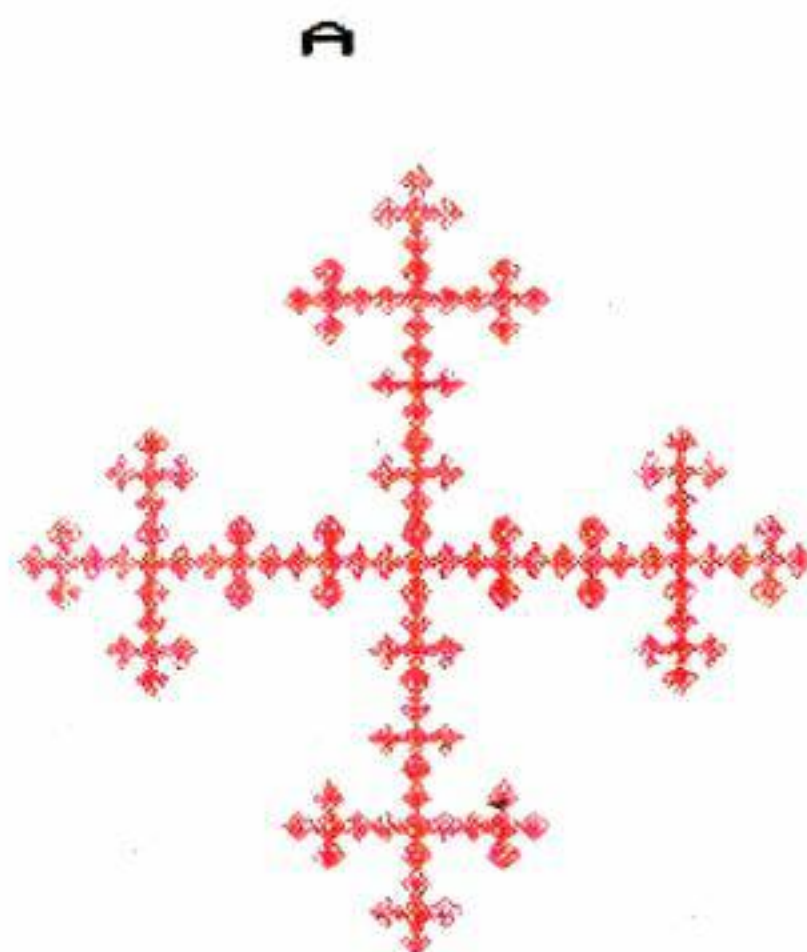
Per capire meglio la spiegazione che seguirà guardate la figura "creazione di una curva di Peano", nella quale sono illustrati i risultati dei vari passaggi.

Immaginiamo di partire da un segmento (ordine 0) e di applicare a questo una trasformazione "quadrata" nel centro.

Cancelliamo, cioè, il terzo centrale del segmento e lo sostituiamo con un quadrato di cui non tracciamo il lato inferiore (ordine 1).

Consideriamo ora ognuno dei 5 segmenti risultanti e su ciascuno applichiamo la stessa trasformazione (ordine 2).

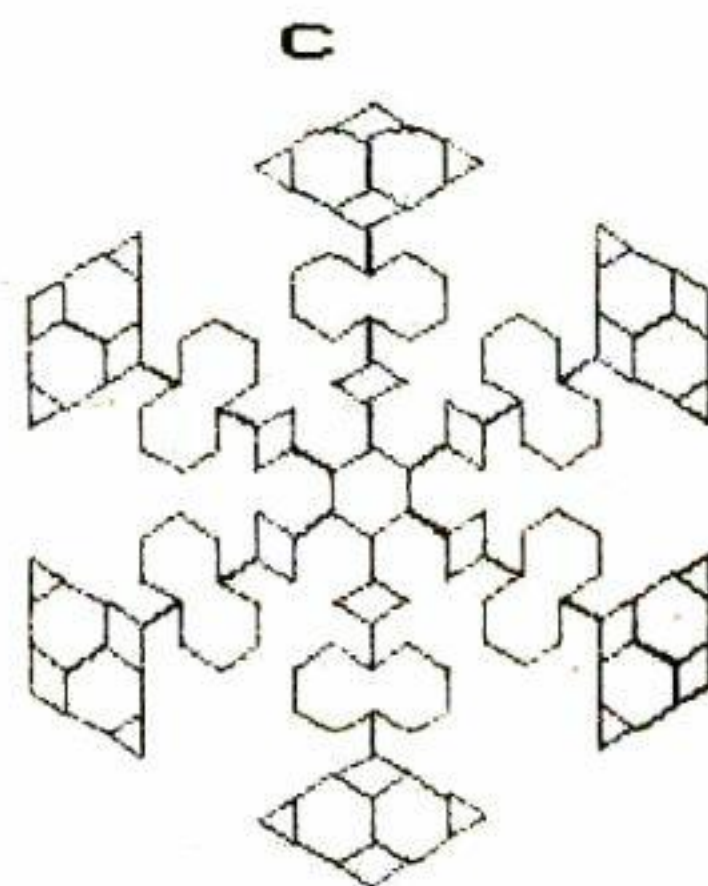
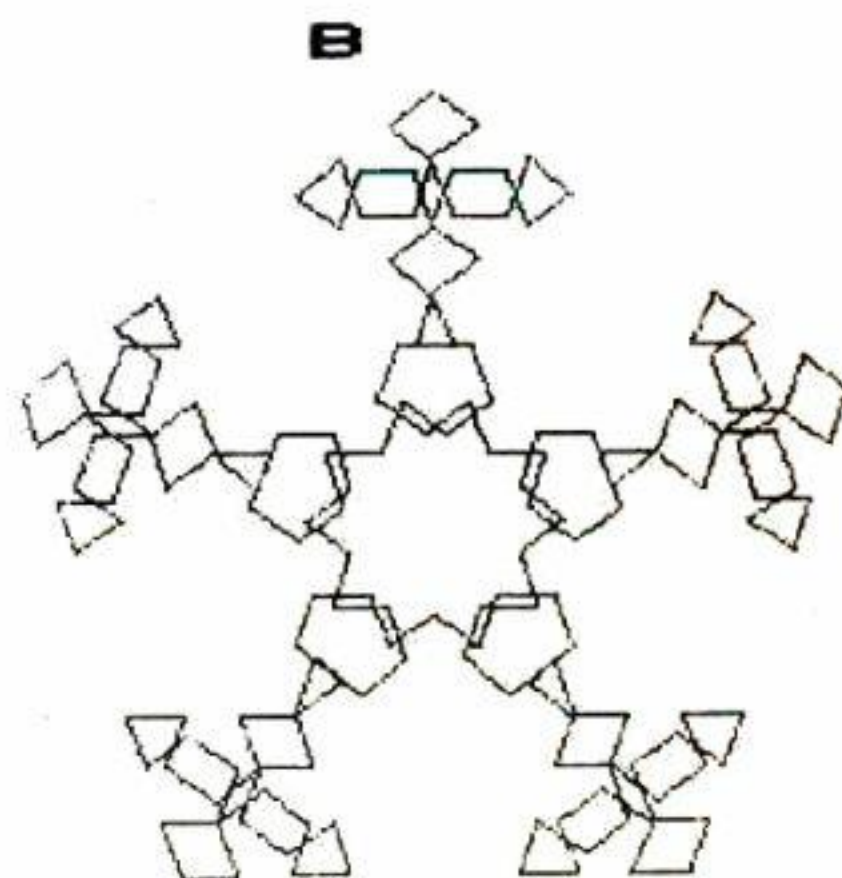
Ripetiamo l'operazione sui 25 lati ri-

A

Provate a digitare una terna dei seguenti valori riportati in tabella. Stupende figure geometriche verranno riprodotte sullo schermo del vostro TV. Come esercizio, provate a indovinare quale terna di valori dà origine alla fig. H, poi inviateci la soluzione.

	Tipo	N. curve	Ordine
A	4	4	4
B	5	5	2
C	6	6	2
D	6	6	3
E	5	6	3
F	3	4	4
G	5	5	4
H	?	?	?
I	4	5	3
L	4	5	5

Ecco alcuni esempi di figure (contraddistinte da lettere dell'alfabeto) che potete realizzare col programma di queste pagine. Alcune sembrano dei veri e propri fiocchi di neve!!

B

C

Infatti le curve di Peano richiedono, per la loro costruzione, l'utilizzo di tecniche ricorsive che non sono prerogativa del BASIC.

Nonostante ciò potrete constatare, guardando le curve realizzate, che anche con tale linguaggio si possono simulare tali procedure, anche se non all'infinito, e quindi realizzare spettacolari "merletti" matematici.

Una doverosa precisazione: per disegnare le curve è indispensabile caricare ed attivare le Routine Grafiche pubblicate sul N.14 di C.C.C.

Gerarchie

Vediamo ora la logica di costruzione della famiglia di curve prese in considerazione.

sultanti (ordine 3) e poi sui 125 nuovi e sempre più piccoli segmenti (ordine 4).

Applicando questo procedimento all'infinito si ottiene una curva di Peano.

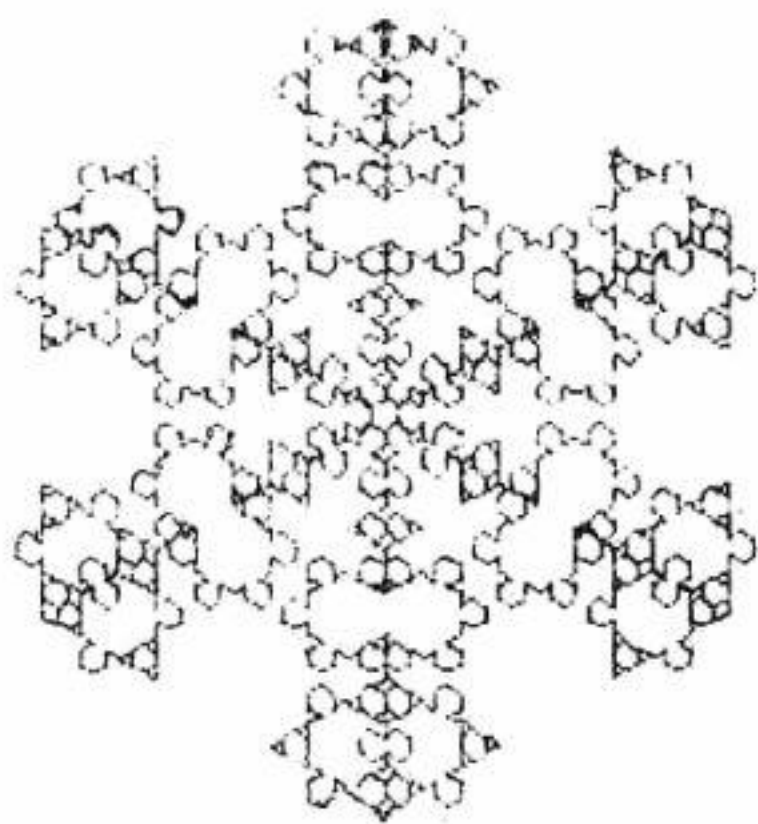
Facile vero? Il problema però è implementare tale algoritmo in BASIC.

Stabilire la lunghezza del lato dell'ennesimo ordine è banale: se L0 è il lato di ordine 0, L=L0/(3↑N).

Le difficoltà nascono riguardo alla direzione in cui deve procedere il nostro cursore ideale.

Nel primo ordine non ci sono problemi: ipotizzando che parta da sinistra dovrà prima "girare" a sinistra (S), quindi due volte a destra (D) ed infine a sinistra (S):

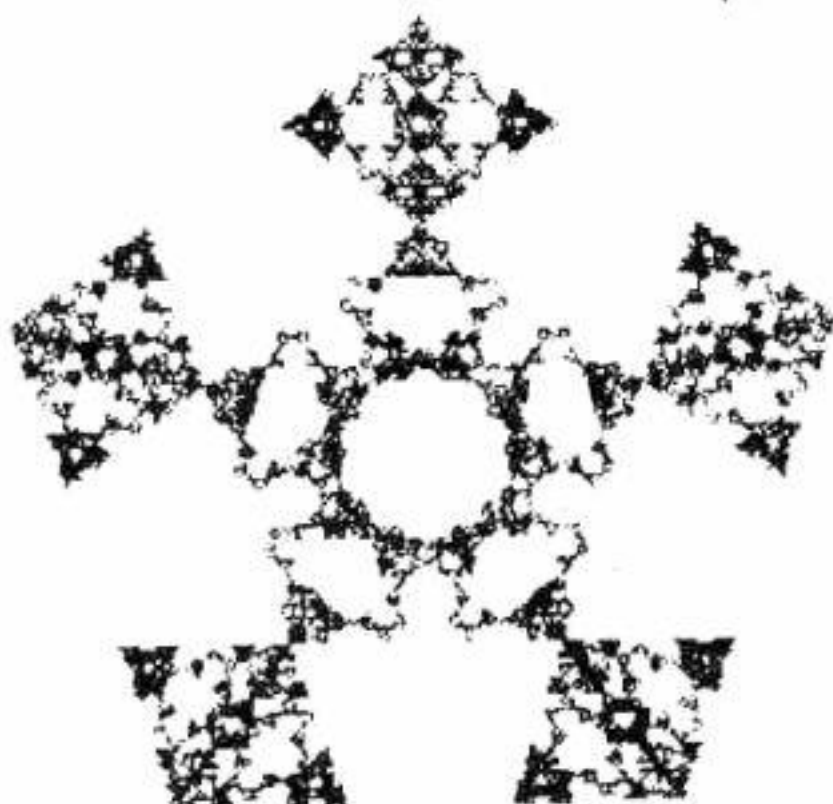
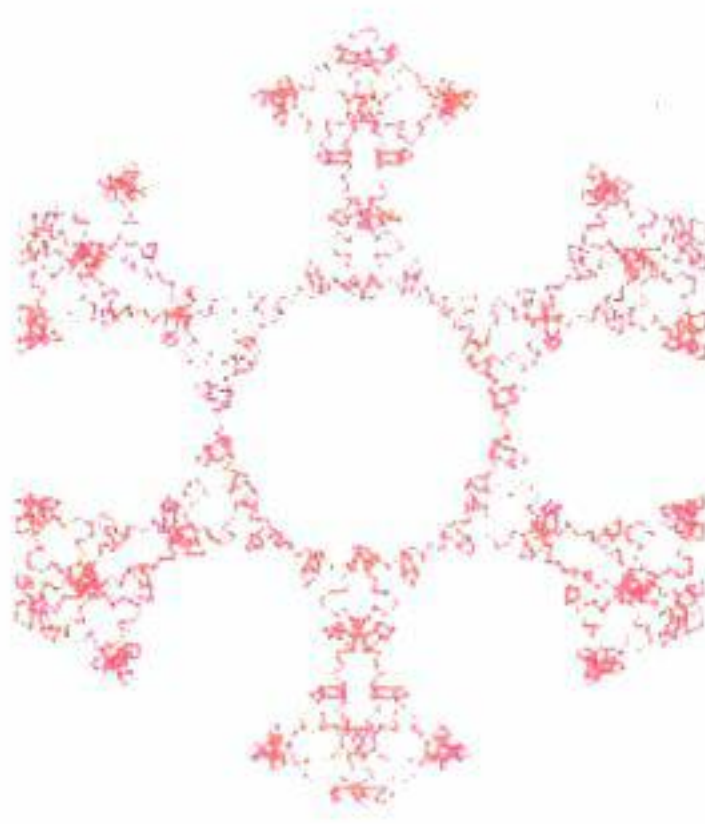
Sequenza 1=(S-D-D-S).

Nel secondo ordine si ripeterà la sequenza 1 per ognuno dei 5 lati e quindi,

sempre partendo dall'estremo sinistro, il percorso sarà:

Sequenza 2=(sequenza 1)-S-(sequenza 1)-D-(sequenza 1)-D-(sequenza 1)-S

Avrete notato che la nuova sequenza è simile alla prima!

Si procede di questo passo all'infinito:
Sequenza N=(sequenza N-1)-S-(sequenza N-1)-D-(sequenza N-1)-D-(sequenza N-1)-S
in cui ogni sequenza è determinata dalla precedente e a sua volta determina la successiva.

Per riuscire a sapere in ogni momento in quale direzione procedere occorre (secondo la soluzione da me individuata) avere un indice per ogni sequenza (dalla sequenza N alla sequenza 1) che segnali quale sarà la mossa successiva.

Per aiutarvi a capire, prendiamo come esempio la numerazione decimale.

Ad ogni cifra da uno a nove possiamo associare una direzione, che chiameremo (D1)...(D9), ottenendo una sequenza simile a quelle illustrate precedentemente.

Vogliamo, partendo per esempio dal numero 000, arrivare a 999 passando da tutti i numeri intermedi.

Incrementiamo le unità: da 001 fino a 009. Ciò equivale ad eseguire la serie di rotazioni predeterminata.

Se ad ogni rotazione facciamo seguire un avanzamento otterremo un percorso.

Incrementiamo ancora: arriviamo a 010, cioè otteniamo ancora una rotazione di tipo (D1) e un ulteriore avanzamento.

Riprendiamo ad incrementare: da 011 a 019, ottenendo un nuovo percorso.

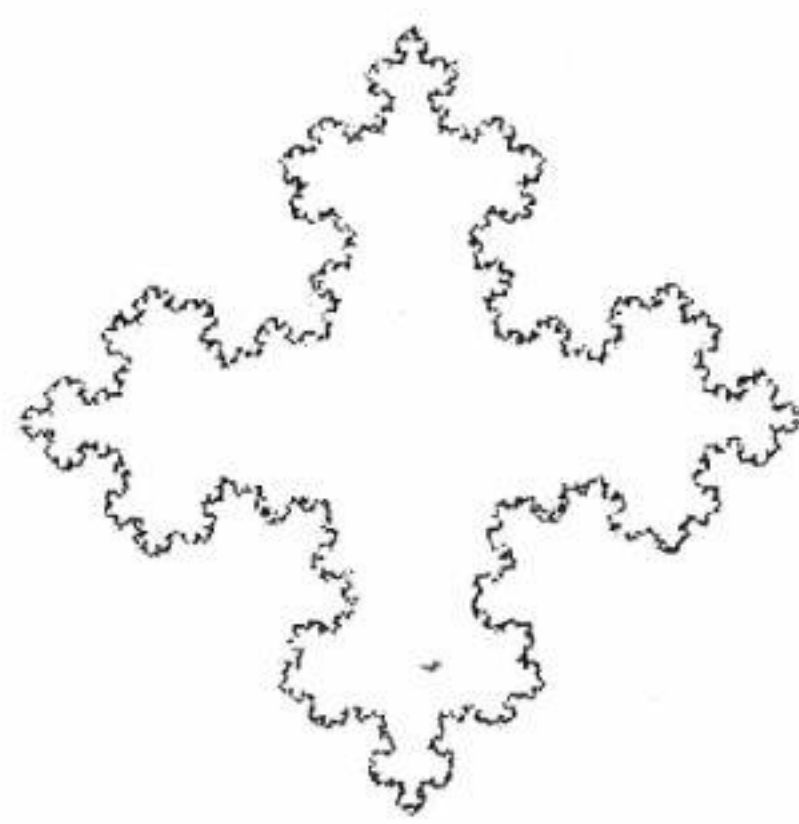
Il passo successivo porterà il numero a 020 con relativa rotazione (D2) e avanzamento.

Si procede in questo modo fino a 099. A questo punto appare oramai chiaro che si sono percorse dieci sequenze di "unità" e si è completata una sequenza di "decine".

Con un ulteriore incremento arriviamo a 100 cioè iniziamo, con (D1), la sequenza delle "centinaia".

Quindi ricominciamo a procedere nel modo descritto.

Arrivati a 999 avremo completato 10↑2=100 sequenze di "unita", 10↑1=10 sequenze di "decine" e 10↑0=1 sequenze di "centinaia".

Ma, ciò che più conta, avremo percorso una dannata curva ricorsiva di ordine 3.

In questo caso i tre indici che hanno permesso di sapere in ogni momento la mossa da effettuare, sono stati le unità, le decine e le centinaia.

Possiamo però ricorrere anche agli elementi di un vettore, così da non essere limitati a sequenze composte da nove movimenti.

Spero che quanto detto sia risultato comprensibile, ma se avete difficoltà, ricostruite il percorso con carta e matita e vedrete che risulterà più chiaro.

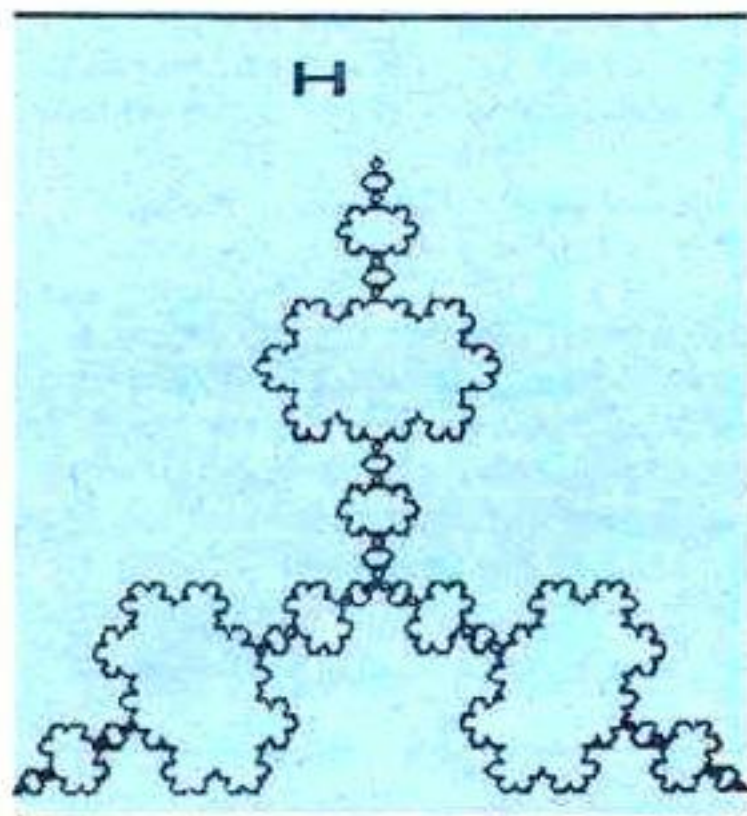
In ogni caso, per ottenere disegni come quelli presentati nelle figure non è necessario sapere come funziona la procedura.

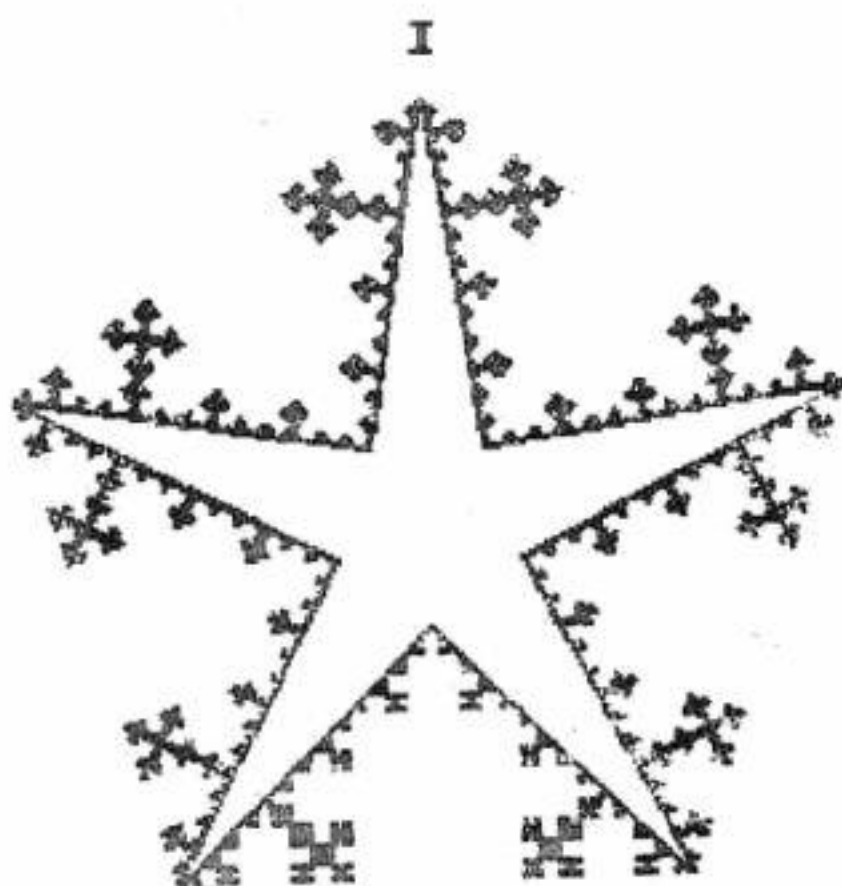
Il programma

Permetterà di creare curve diverse ma basate tutte sullo stesso principio base illustrato.

Ricordate, prima di farlo girare, di caricare le Routine Grafiche del N. 14 di C.C.C. senza le quali non può funzionare.

Dato il RUN vi verrà chiesto di specificare 3 informazioni.

1/ Tipo di trasformazione: potete scegliere dal triangolo al decagono. Digitate quindi la cifra corrispondente al numero di lati (da 3 a 10). Volendo trasformazioni con poligoni di 11 o più lati basta inserire la linea
205 DIM D(nn)
dove nn è il numero massimo di lati consentiti.

2/ Numero di curve: per ottenere delle figure chiuse occorre unire tra loro più curve uguali, quindi risponde con un numero da 3 in su. Potete anche superare il limite di 10 ma, generalmente, non si ottengono disegni molto belli con più di 8 curve.

3/ Numero d'ordine: per ultimo va specificato quante volte va applicata la trasformazione e ciò, come avrete appreso leggendo l'articolo, viene determinato dall'ordine della ricorsione. Più grande è il numero che impostate, più intricata sarà la curva ma più tempo occorrerà per disegnarla. Valori consentiti tra 0 e 10. Anche in questo caso si può aumentare il limite dimensionando il vettore O(nn). Comunque, data la risoluzione di 320*200 dello schermo, è inutile spingersi oltre il quinto o il sesto ordine.

Le tre informazioni vanno separate dalla virgola.

Quindi battete RETURN e inizierà la "creazione".

Quando il disegno sarà terminato il programma si fermerà per consentirvi di ammirare il risultato e aspetterà che premiate un tasto. Chiederà quindi se volete creare un'altra curva.

Nelle didascalie delle figure pubblicate sono indicati gli input che hanno generato le relative curve: potranno servirvi da esempio e per controllare che il programma funzioni regolarmente.

Se avete una routine di hard-copy adatta alla vostra stampante, potete inserirla nel listato e fare stampare le figure più belle.

A questo proposito ricordo che sul numero 18 di C.C.C. ne è stata pubblicata una per la MPS 802, mentre sul numero 10 di Commodore potete trovare quella per la MPS 801.

Casi particolari

Alcune realizzazioni potranno eccédere i limiti di schermo nonostante la parametrizzazione della lunghezza del lato base.

In tal caso, modificate nella linea 330 il valore 199 (diminuendolo).

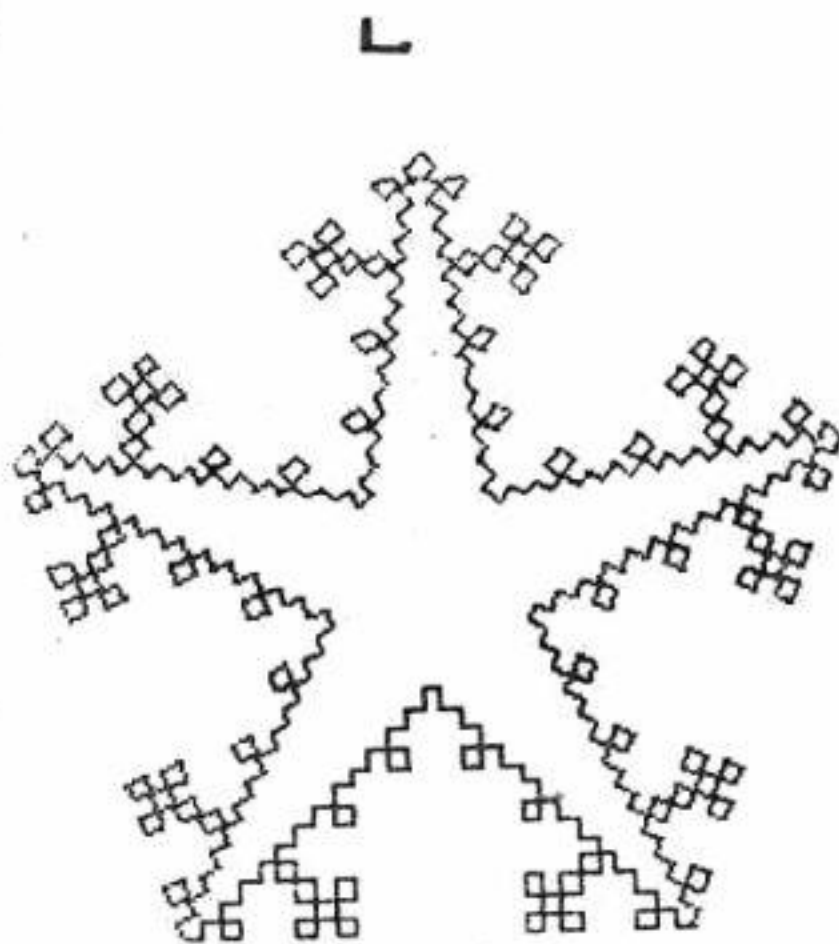
Nella linea 320 invece sono definite le coordinate del punto da cui comincia il disegno e l'orientamento iniziale del "cursore" (dipendente dal numero di curve da tracciare).

Volendo ottenere figure diverse (con trasformazione "esterna" anzichè "interna") eliminate il segno meno nell'istruzione GR=-2* /P (linea 250).

Nel vettore D() vengono conservati gli angoli di rotazione della sequenza di trasformazione (linee 260-300). Il primo e l'ultimo sono sempre uguali e di segno opposto agli intermedi. La somma totale deve essere pari a zero, così da poter ritornare alla direzione iniziale alla fine di ogni sequenza .

Il resto del breve listato che fa da contorno alla routine principale è facilmente comprensibile quindi è superfluo parlarne.

La parte più ostica va dalla linea 590 alla linea 640 ed implementa l'algoritmo che ho cercato di spiegare con l'esempio dei numeri decimali.

Qui al posto di unità, decine, ecc., abbiamo O(1), O(2), ecc.

P (tipo di trasformazione) determina quando una sequenza è stata completata e bisogna passare al punto successivo di quella di ordine superiore.

Dalla linea 660 alla linea 690 troviamo la routine che fa avanzare il cursore nella direzione predeterminata.

Spero con ciò di avere spiegato tutto e di non avere detto qualche assurdità parlando di curve a me quasi sconosciute.

In caso contrario godetevi il programma che, come direbbero i cinesi, vale più di mille parole.

Attenzione!!!

Al lettore che per primo invierà i tre parametri che consentono di disegnare la figura che in queste pagine è contrassegnata con la lettera "H", la Redazione invierà gratuitamente un pacco contenente libri, arretrati delle riviste, eccetera.

Ricordatevi di indicare chiaramente, oltre ai tra parametri richiesti, il vostro nome cognome ed indirizzo.

SYSTEMS EDITORIALE
Commodore Computer Club
Rubrica "Curve di Peano"
Viale Famagosta, 75
20142 MILANO

```
100 REM GRAFICA PER COMMODORE 64
110 :
120 REM CURVE DI PEANO
130 :
140 REM CARICARE ROUTINE GRAFICHE
150 REM PUBBLICATE SUL N.14 DI C.C.C.
160 :
170 REM SOFTWARE DI DANILO TOMA
180 :
190 PRINTCHR$(147)
200 POKE53280,6
210 PRINT"TIPO DI TRASF., N. DI CURVE, N. D'ORDINE"
220 INPUTP,S,N
230 PRINT
240 IFW=0THEN:←CLEAR:W=1
245 ←GRAF6,1:←COLOR1
250 GR=-2*π/P:R=-2*π/S
260 FORI=2TOP-1
270 D(I)=-GR
280 NEXT
290 D(1)=GR*(P-2)/2
300 D(P)=D(1)
310 :
320 X1=0:Y1=99:AN=R/2+π
330 L=199*SIN(R/2)/3↑N
340 :
350 FORI=1TOS :REM NUMERO DI CURVE
360 :
370 FORK=1TON
380 O(K)=0 :REM INIZIALIZZA INDICI
390 NEXT
400 :
410 GOSUB660 :REM ELABORA
420 :
430 AN=AN+R :REM DIREZIONE DEL
440 NEXT     :REM PROSSIMO LATO
450 :
460 REM PREMI UN TASTO
470 WAIT198,1:GETA$
480 :
490 ←TEXT6,14
500 PRINT"ANCORA ? (S/N)"
510 WAIT198,1:GETA$
520 IFA$="N"THENEND
530 PRINT"CANCELLO SCHERMO HI-RES?(S/N)":WAIT198,1:GETA$
540 W=0:IFA$="N"THENW=1
550 GOTO200
560 REM ELABORA E DISEGNA
570 REM LA CURVA
580 :
590 K=0
600 K=K+1
610 IFK>NTHENRETURN
620 IFO(K)=PTHENO(K)=0:GOTO600
630 O(K)=O(K)+1
640 AN=AN+D(O(K))
650 :
660 X2=X1+COS(AN)*L
670 Y2=Y1+SIN(AN)*L
680 ←DRAWX1,Y1,0,X2,Y2,0
690 X1=X2:Y1=Y2
700 GOTO590
```

Imparando ad imparare

Una stellina passo dopo passo

In quest'ultima puntata impariamo a verificare un programma di E.Li.Ana., e a far prendere decisioni alla coccinella.

di D. Matturro & M. L. Nitti

Spesso e volentieri restiamo perplessi dopo aver eseguito il nostro programma, perchè non abbiamo ottenuto quello che realmente volevamo. Dopo aver dato il comando ESEGUI, ci ritroviamo a vedere sullo schermo la coccinella che si muove in un modo che non ci sembra "giusto" (ma questo calcolatore non capisce proprio niente ?!) oppure vediamo risultati numerici che non ci soddisfano. Questo chiaramente può succedere con qualsiasi linguaggio. In termine tecnico, l'operazione di verifica di un programma si chiama "debug". A seconda del tipo di calcolatore usato e del tipo di linguaggio, esistono programmi più o meno sofisticati per compiere questa operazione.

Per poter capire più facilmente dove abbiamo sbagliato, E.Li.Ana ci viene in aiuto con il comando diretto:

PASSO-PASSO

Mediante questo comando, impartito prima dell'esecuzione, vedremo nella parte più alta dello schermo l'istruzione che in quel momento viene eseguita. Per continuare nel lavoro, sarà necessario premere ogni volta la barra spaziatrice. In questo modo si può osservare lo schermo quanto tempo si desidera, e individuare quindi molto chiaramente quale "passo" o linea di programma, dà luogo a risultati indesiderati.

Per utilizzare tale possibilità dovrete quindi, una volta scritto il programma, dare il comando PASSO-PASSO seguito da ESEGUI. Per interrompere questa particolare esecuzione e tornare alla pagina di QUADERNO, basta premere il tasto RUN/STOP.

Immaginate di voler disegnare una stellina. Come si può procedere?

Scriviamo anzitutto le linee che permettono di disegnare una punta di lunghezza L:

```
 10 ALBUM
 15 CESTINO
 20 MATITA
 25 VARIABILE L=10
200 AVANTI L
210 DESTRA 30
220 INDIETRO L
230 FINE
```

Diamo ora i due comandi PASSO-PASSO e ESEGUI. Osservato il risultato sullo schermo, vediamo che è necessario far voltare a sinistra la coccinella per disegnare una seconda punta.

Aggiungiamo perciò:

```
 30 SOTTOPROGRAMMA 300;2
100 LINEA 100
300 SOTTOPROGRAMMA 200;1
310 SINISTRA 90
320 TORNA
```

modificando la linea 230 in

```
230 TORNA
```

In questo modo usiamo le linee 200-230 come sottoprogramma che genera una punta, la linea 300 come sottoprogramma che genera la punta e la rotazione in vista della punta successiva.

Ora è sufficiente dare il comando ESEGUI per avere l'opzione passo per passo, che rimane abilitata. Osservate che alla fine la linea 100 rimane in esecuzione finchè non premete RUN/STOP.

Benissimo! Tutto funziona. Ora però, prima di fare la prova successiva, diamo il comando

NORMALE

per eliminare l'opzione PASSO-PASSO. Resta ora da capire con quante punte sarà completata la stellina.

Proviamo a sostituire, nella linea 30, il parametro di chiamata 2 con il valore 6:

30 SOTTOPROGRAMMA 300;6

aggiungendo inoltre

35 VOLA
40 AVANTI 30

per allontanare la coccinella dalla stellina e poter vedere meglio l'effetto finale. Vedrete formarsi una stellina a sei punte proprio perfetta. Notate che, se nella linea 25 modificate il valore di L, potrete disegnare stelle di grandezza diversa. A questo punto potete sbizzarrirvi nella creazione di stelline più o meno grandi, ed anche con un numero diverso di punte, se modificate l'angolazione prodotta dalla linea 310.

Ecco qui il listato completo, per chi desidera vedere subito il risultato senza seguire in dettaglio tutte le spiegazioni:

Stellina a sei punte

```
 10 ALBUM
 15 CESTINO
 20 MATITA
 25 VARIABILE L=10
 30 SOTTOPROGRAMMA 300;6
 35 VOLA
 40 AVANTI 30
100 LINEA 100
200 AVANTI L
210 DESTRA 30
220 INDIETRO L
230 TORNA
300 SOTTOPROGRAMMA 200;1
310 SINISTRA 90
320 TORNA
```

Decisioni nel programma

A volte scrivendo un programma si desidera poter prendere strade diverse, anzichè seguire una sola linea di condotta. E.Li.Ana. ci propone per questo scopo l'istruzione SE, che ha la seguente forma:

(linea) SE (condizione);
ISTRUZIONE1
(linea
successiva) ISTRUZIONE2

Il risultato è il seguente: se la condizione è vera, si passa ad eseguire l'istruzione 1 e poi in sequenza l'istruzione 2; se la condizione è falsa, l'istruzione 1 non viene eseguita.

Vediamo un semplicissimo esempio, per calcolare il maggiore tra due numeri A e B:

Massimo tra due numeri A,B

```
10 QUADERNO
15 VARIABILE B=10
20 VARIABILE A=5
25 VARIABILE M=A
30 SE B>A; VARIABILE M=B
35 SCRIVI IL MASSIMO TRA "A"
   E "B" E' "M"
40 FINE
```

La variabile M conterrà, alla fine dell'esecuzione, il massimo tra A e B. Si assegna ad M il valore A (linea 25); se vale la condizione B>A ,alla variabile M viene assegnato il valore B (linea 30). Nel caso in cui la condizione non sia vera, l'istruzione VARIABILE M=B non viene eseguita e si passa direttamente alla

stampa dei risultati. Nell'esempio, i valori assegnati ad A e a B sono rispettivamente 5 e 10 e il massimo risulta essere B. Provate ad eseguire il programma con valori differenti.

Vediamo ora un esempio un po' più complicato: il calcolo della potenza N di un numero intero positivo A. Ecco il listato:

Potenza N-ESIMA di A

```
10 QUADERNO
15 VARIABILE A=3
20 VARIABILE N=5
25 VARIABILE P=1
30 SCRIVI CALCOLO
   LA POTENZA
   "N" DI "A"
35 VARIABILE P=P*A
40 VARIABILE N=N-1
45 SE N=0; LINEA 55
50 LINEA 35
55 SCRIVI IL RISULTATO E' "P"
60 FINE
```

Nella linea 35 viene eseguita, passo per passo, la potenza desiderata: all'inizio in P si ha 1, poi A, poi A*A, eccetera; ogno volta che si esegue un prodotto, N viene decrementato di una unità. Nella linea 45 si controlla il valore di N ; se N>0 si torna alla 35 ad eseguire ancora un prodotto; quando N diventa 0, l'esecuzione passa alla linea 55 e il procedimento è concluso.

In questo programmino, nelle linee 35-50 è stato realizzato un "ciclo" o "loop": una serie di istruzioni viene eseguita un certo numero di volte (N in questo caso) finchè si verifica una precisa condizione (nell'esempio: N=0) e il ciclo è concluso. Per chi conosce il basic, la medesima struttura logica è realizzata con l'istruzione FOR seguita da NEXT.

La tela di Penelope

Nella prima puntata di questa rubrica (n. 24 di C.C.Club - ott.85) abbiamo visto, tra le istruzioni di base per la coccinella, anche il comando GOMMA. Utilizzabile da programma, questo comando ordina alla coccinella di cancellare, d'ora in avanti, tutte le linee già tracciate che incontrerà sul suo cammino.

Ora vi proponiamo un semplice listato che mostra come, unendo l'uso di una variabile all'uso dell'istruzione SE, si possa costringere la povera coccinella a disegnare e poi subito dopo cancellare la stellina, senza mai fermarsi.

Disegna e cancella

```
10 ALBUM
15 CESTINO
20 MATITA
25 VARIABILE L=20
30 VARIABILE G=0
35 SOTTOPROGRAMMA 300;6
40 SE G=0; LINEA 60
50 MATITA
55 LINEA 30
60 VARIABILE G=1
65 GOMMA
70 LINEA 35
100 LINEA 100
```

(seguono le linee 200-320 del programma Stellina)

L'idea fondamentale del programma sta nella definizione e modifica della variabile G. In pratica, esistono due "stati" della coccinella: stato G=0, in cui disegna; stato G=1, in cui cancella. Gli stati vengono alternati, assegnando di conseguenza la direttiva GOMMA o MATITA. Questo procedimento può essere esteso ad altri usi, e opportunamente modificato.

Per finire, vi lasciamo con la proposta di realizzare un programmino che, utilizzando l'ormai sfruttatissima stellina a sei punte, costruisca nella pagina di album varie stelline con lati di lunghezza diversa, eventualmente assegnati a caso.

Termina, con questa puntata, la "chiacchierata" sul corretto utilizzo del pogramma E.LI.ANA. pubblicato sulla rivista su cassetta Commodore Club edito dalla nostra stessa casa editrice.

Coloro che intendono procurarsi sia i fascicoli arretrati di C.C.C. che trattano questo argomento, sia il nastro-cassetta contenente il linguaggio E.LI.ANA. possono telefonare in redazione (02-8467348) per uleteriori informazioni.

"Spaghetti Basic"

Impariamo a sbagliare

di **Alessandro de Simone**

Ecco la seconda puntata su come affrontare il nuovo oggetto misterioso che è entrato (o sta per entrare) nella vostra casa: il computer, Commodore, naturalmente la terza ed ultima parte verrà pubblicata nel prossimo fascicolo.

Nel numero scorso avevamo proposto alcuni esercizi utili perfino per chi non possiede un calcolatore.

Ci auguriamo che nel frattempo abbiate provveduto ad acquistare un Commodore e verificato l'esito degli esercizi stessi.

Riportiamo, ad ogni buon conto, sia le domande che le soluzioni.

N.B. Il simbolo (R) sta per "Premere il tasto Return".

Attenzione anche a non confondere il tasto alfabetico "O" col tasto numerico zero (0).

Esercizio		Soluzione
PRINT 123+233+45	(R)	*401*
PRINT 2 * (4+5/2)	(R)	*13*
PRINT (5+7) * (8+3)	(R)	*132*
PRINT SQR (16)	(R)	*4*
PRINT 52	(R)	*52*
PRINT	(R)	*Rigo vuoto*
PRINT 5 SQR(2+7)	(R)	*5 3*

Problemi:

- Calcolare l'area di un quadrato di latot8:
PRINT 8*8 (R) *64*

- Calcolarne il perimetro:
PRINT 4*8 (R) *32*

- Area di un triangolo rettangolo isoscele con cateto=10
PRINT 10*10/2 (R) *50*

- Calcolare l'ipotenusa del triangolo precedente:
PRINT SQR(10↑2+10↑2) (R) *14.1421...*

- Calcolare la circonferenza di raggio
10:PRINT 2*3.1415*10 (R) *62.8318...*

- Calcolare l'area del cerchio precedente:
PRINT 3.1415*10*10 (R) 314.1592...

* Calcolare l'I.V.A. (18%) da pagare sulla cifra di 100000
PRINT 100000*18/100 (R) *18000*

Per usare con padronanza un qualsiasi oggetto è necessario utilizzarlo cercando all'inizio di sbagliare a bella posta il più spessò possibile in modo da individuare più facilmente, durante il suo uso normale, eventuali disfunzioni. L'esperienza accumulata in casi semplici sarà preziosissima per riconoscere inevitabili errori di battitura come pure incongruenze di calcolo.
La stessa cosa vale per chi legge queste pagine: proviamo a vedere che succede se digitiamo in modo errato i semplici comandi appresi finora.

Invece di digitare:

PRINT 123 + 456 (R)

digitiamo:

PRINT 123 +

ed ovviamente il tasto Return (non dimenticate *mai* di premere Return altrimenti il computer non può eseguire l'ordine).

Notiamo con sorpresa che la somma (ma quale somma, del resto?) non è stata eseguita e che viene visualizzato al suo posto il messaggio di errore:

?SYNTAX ERROR
READY.

Esaminiamo in dettaglio ciò che è accaduto:

1/ Quando abbiamo premuto il tasto di ritorno carrello dopo aver battuto il tasto "+", il computer ha cercato di "interpretare" ciò che avevamo digitato.

Il primo carattere che incontra è P seguito da R,I,N,T. La parola PRINT, che appartiene al "vocabolario" (elenco di parole chiave) in possesso del computer, rappresenta un preciso comando di stampa e significa esattamente: "Preparati a visualizzare sullo schermo del computer i dati, opportunamente elaborati, che tra breve incontrerai".

2/ I caratteri presenti dopo PRINT sono, nel caso esaminato, numerici (123). Il Sistema Operativo, a questo punto, sa che gli verrà tra breve richiesto di "trattare" il valore 123. Una conferma di cio' la trova nell'interpretare il carattere successivo, al segno più (+), e si predispone ad effettuare una somma con il valore numerico che, non ha dubbi, incontrerà tra breve, ma...

3/ ... sorpresa: il computer non trova alcun carattere e ne deduce, nella sua infinita sapienza, che l'operatore si è distratto o che gli ha voluto fare uno scherzo. Nel dubbio, il calcolo non viene eseguito e viene visualizzato un discretissimo messaggio di errore fortunatamente privo di imbarazzanti commenti sulle facoltà intellettive dell'utilizzatore.

4/ A questo punto il computer, per non appesantire l'atmosfera, comunica che è pronto (READY.) ad accettare altri ordini e fa lampeggiare il cursore come se nulla fosse accaduto.

L'esperienza effettuata dimostra diverse cose utili:

- *Il linguaggio usato dal computer è un linguaggio interprete*. Ciò vuol dire che quando chiediamo di eseguire un'operazione (servendoci, ripetiamo, del tasto Return), il Sistema Operativo legge ca-

rattere per carattere l'ordine stesso e *solo se* tutto risulta in regola esso viene eseguito.

Se invece l'ordine richiesto non rispetta i canoni stabiliti, l'elaborazione in corso viene interrotta ed un messaggio viene emesso come avvertimento. Tocca all'operatore indovinare il perchè della segnalazione e porvi rimedio.

• Nulla può essere sottinteso. In caso di somma di due addendi (=numeri da sommare tra loro) di cui uno nullo, questo deve essere esplicitamente indicato con 0 (attenzione a non confondere il tasto numerico zero "0" col tasto alfabetico "O"). Esempio:

PRINT 123 +0 (R)
PRINT 4+0+5+0 (R)

Si ricorda che il segno della moltiplicazione (per) nei calcolatori è sostituito dall'asterisco (*). Per moltiplicare 10 per 2, ad esempio, si ricorre alla simbologia:

PRINT 10 * 2.

Commettiamo ora un altro errore:

PRONT 123 + 456 (R)

Anche in questo caso viene emessa la stessa segnalazione di errore. Stavolta, però, gli addendi sono stati scritti in modo corretto. Errato è invece l'ordine (PRONT al posto di PRINT) che non viene ovviamente considerato esistente nella tabella delle parole-chiave che il computer ha, comunque, scrupolosamente esaminato dalla prima all'ultima.

In conclusione è sufficiente un solo sbaglio nella battitura per fare apparire messaggi di errore.

Alcuni computer (come il C16, il PLUS 4, il C128) segnalano il punto esatto in cui viene riscontrata l'imprecisione facendo lampeggiare il comando che ha dato origine alla segnalazione. Il C64 ed il Vic 20, purtroppo, non possiedono il tasto HELP che consente questa facilitazione ed è quindi compito dell'utilizzatore rintracciare il motivo della mancata esecuzione dell'ordine.

Operazioni contemporanee

Se vogliamo eseguire due somme non è necessario eseguire prima l'una e poi l'altra. Le due richieste di calcolo possono essere poste sullo stesso rigo, a patto, ovviamente, di seguire alcune semplici regole di "sintassi" (non dimentichiamo che stiamo utilizzando un linguaggio, il basic, dotato di regole sintattiche come qualsiasi altra lingua). Nella maggior parte dei computer in commercio due ordini diversi sono accettati purchè vengano separati dal carattere di doppio punto (:). E' bene notare, quindi, che questo carattere, contrariamente a quanto sembri a prima vista, assume un significato completamente diverso da quello, comunemente noto, di "diviso" utilizzato come simbolo della divisione.

Esempio:

PRINT 123+456:PRINT 789+122 (R)

Questo modo di operare consente di risparmiare spazio sul video del computer oppure, in alcuni casi che esamineremo, di premere un minor numero di tasti. Si badi bene che talvolta a causa del numero limitato di caratteri per ciascuna riga, è probabile che, nel digitare più ordini concatenati, alcuni di essi risultino interrotti su di una riga di schermo e continuino su quella successiva, come ad esempio (caso del Vic 20):.

```
PRINT 123456 +7890:PR
INT 100 + 478 (R)
```

(Il primo rigo contiene 22 caratteri, compresi gli spazi, ed il secondo i rimanenti).

Nel caso invece di un computer da 40 colonne come il Commodore 64, si può avere un'analoga difficoltà come nell'esempio che segue:

```
PRINT 1234 +5678:
PRINT SQR(12*44): PR
INT 456/789 (R)
```

Nei casi appena visti l'insolita divisione della parola PRINT o di altre... "cose" (come numeri e comandi), non deve essere considerata come una imprecisione che può causare errori di interpretazione da parte del computer. Ciascun computer, infatti, "vede" una serie di caratteri come un'unica lunga linea. Nel caso del Vic 20 è possibile digitare fino ad 88 caratteri consecutivi prima di battere il tasto di ritorno carrello (in caso contrario viene emessa una segnalazione di errore). Nel Commodore 64 i caratteri accettati sono 80 che corrispondono a due righe consecutive (il video è del formato 25 linee di 40 caratteri ciascuna). Tutti gli apparecchi considerano, in ogni caso, la fila di caratteri da "trattare" come un'unica riga la cui lunghezza viene misurata dal primo all'ultimo carattere battuto. Ciò che appare sul video, e le eventuali insolite divisioni di parole e comandi, danno fastidio solo... all'utilizzatore: il Sistema Operativo non le nota affatto.

Dimostriamo ulteriormente che il basic è un linguaggio interprete. Battiamo:

PRINT 123+456: PRONT 2+2 (R)

Otteniamo come risposta:

579
?SYNTAX ERROR
READY.

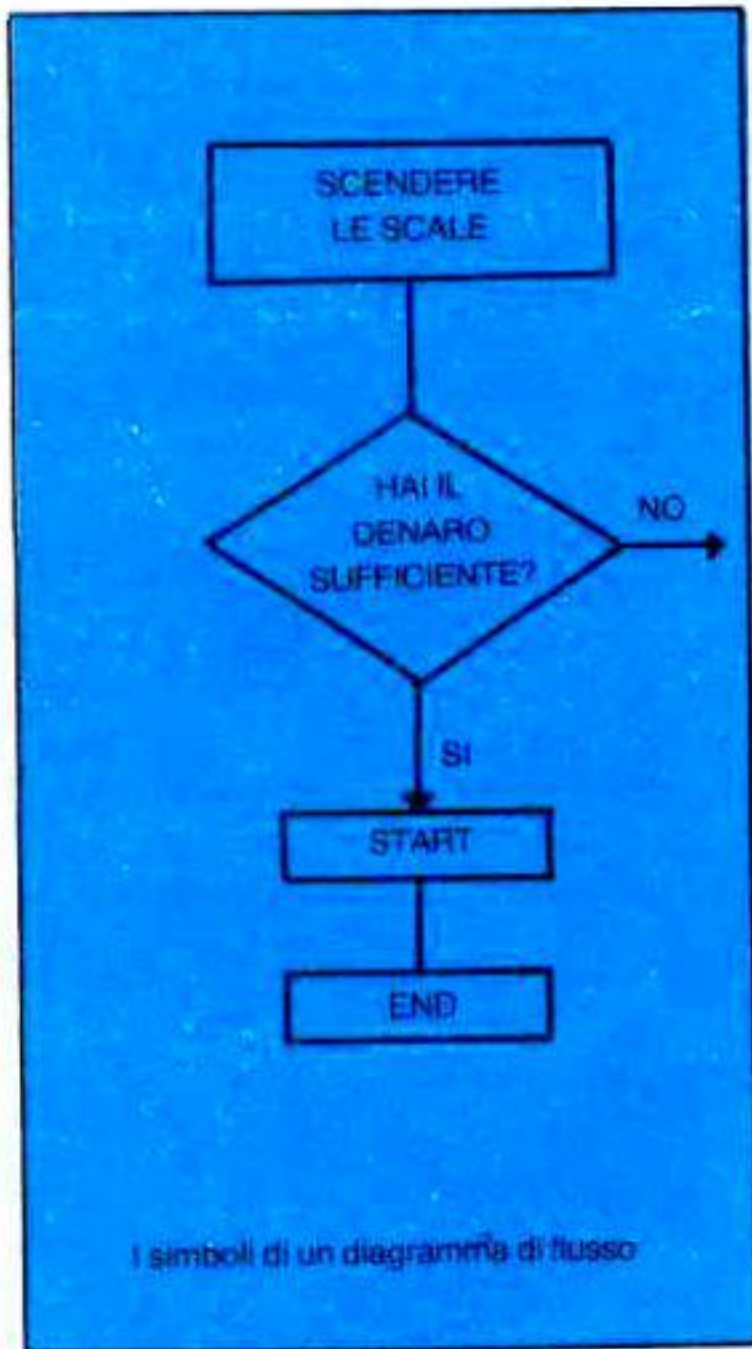
Il computer, infatti, legge ed interpreta, carattere per carattere, PRINT 123 + 456. Dopo quest'ultimo valore incontra il carattere due punti, deduce che l'istruzione appena incontrata termina in questo punto e la esegue senza indugio dando come risultato la visualizzazione del valore-somma 579.

Subito dopo si appresta ad interpretare le parole chiave successive al doppio punto (che ricordiamo, *non* significa "diviso") ma incontra alcune difficoltà (PRONT invece di PRINT), interrompe il lavoro ed emette il messaggio.

Il linguaggio, dunque, lavora fino a che non incontra incongruenze tra ciò che gli viene richiesto di interpretare o elaborare e ciò che invece gli è stato "insegnato", vale a dire le rigide regole sintattiche memorizzate da qualche parte all'interno del computer e sempre controllate al momento di eseguire ordini.

Abbreviazioni possibili

I Commodore accettano "abbreviazioni" di comandi lunghi in modo da evitare all'utilizzatore la fatica di battere molti tasti. Il comando più noto è proprio PRINT che può essere sostituito dal punto interrogativo.

I simboli di un diagramma di flusso

Ad esempio invece di battere:
PRINT 2+2 (R)

è sufficiente digitare:
? 2+2 (R)

A questo punto, per consentire al lettore che ha appena aquistato un personal computer, come a quello che... lo sogna ancora, proponiamo una serie di esercizi da eseguire dapprima con carta e penna e poi col calcolatore per la verifica:

Che cosa accade se digitiamo:.

a) PRINT 123 (R)
b) PRINT 123*456: (R)
c) ? 123:::PRINT123+200 (R)
d) PRINT 123.54 *2 (R)
e) PRINT98-251:?:PRIN 3+98 (R)
f) PRINT 22+45*27/12-73 (R)
g) PRINT (22+45)*27/(12-73) (R)

Risposte

Ecco che cosa "pensa" il computer dopo che premiamo il tasto Return nei vari esercizi proposti:

a/ Il computer segue il ragionamento: "Mi è stato ordinato di scrivere (PRINT) il numero 123; poichè la riga termina qui (R) eseguo la visualizzazione di 123 e mi metto a disposizione per altri ordini".

b/ "Scrivo il risultato del prodotto (*) di 123 x 456. I due punti che incontro subito dopo mi indicano che *dovrebbe* esserci un altro ordine da eseguire. Ma... non trovo null'altro che il carattere di termine riga (R). Termino dunque di operare *senza* emettere un messaggio di errore dato che errori non vi sono."

c/ "Scrivo il numero 123. Il gruppo di tre doppi punti valgono come se fossero uno solo. Al rigo *successivo* scrivo il risultato della somma 123+200".

d/ "Il numero degli spazi dopo l'ordine non mi fanno nè caldo nè freddo: eseguo l'ordine come se le cifre fossero ravvicinate. L'unica differenza consiste nel fatto che mi hanno fatto perdere un pò di tempo per riconoscere (e scartare) i caratteri che non hanno senso (spazi bianchi detti anche "blank"). Considerando che però viaggio alla velocità di un milione di operazioni al secondo..."

e/ "Visualizzo la differenza 98-251. Al rigo successivo scrivo... nulla, e mi blocco al terzo ordine: Che diavolo vorrà dire PRIN ? Ma guarda un po' che mi doveva capitare!"

f/ "Finalmente qualcosa di meno elementare! Dunque... eseguo dapprima la moltiplicazione, poi la divisione e infine le somme algebriche a meno che...".

g/ ... non vi siano le parentesi. Il loro contenuto va risolto con assoluta priorità rispetto alle cifre presenti al di fuori di esse. Un controllo però va fatto prima di iniziare il calcolo: vediamo... una, due, parentesi aperte... una, due, parentesi chiuse. Tutto a posto: il numero delle parentesi aperte è eguale a quello delle chiuse. Eseguo correttamente il calcolo e, almeno per oggi, basta con i SYNTAX ERROR.

Provate ad alterare il numero di parentesi aperte o chiuse oppure ad inserire il carattere di doppio punto (:) dove non serve: imparerete in tal modo altri casi in cui è possibile ottenere segnalazioni di errore.

Livelli di parentesi

A scuola abbiamo imparato a servirci di tre livelli di parentesi (Graffe, Quadre, Tonde). Questo modo di operare è però utile solo per facilitare la "lettura" dell'espressione matematica all'individuo... umano che la legge.

Il computer non ha difficoltà ad interpretare il corretto odine di lettura.

Esempio:

a/ A scuola...

$$4 \cdot \left\{ 5+2 \left[\frac{(3+2)}{4} - \frac{7}{6} \right] \right\}$$

b/ ...col personal computer...

PRINT 4*(5+2*((3+2)/4-7/6))

Come è facile notare c'è indubbiamente, da parte di noi... umani, una difficoltà di "lettura" nel secondo modo rispetto al primo.

I tre tipi di parentesi e la scrittura sotto forma di frazioni sono indubbiamente più semplici se scritte come abbiamo imparato a scuola. Purtroppo non c'è altro modo di esprimere l'operazione da effettuare nel caso di calcoli a più livelli di parentesi. Si rassicuri, comunque, il lettore: è solo questione di abi tudine e di esercizio; dopo un pò di tempo scomparirà ogni difficoltà sia nell'impostare un'espressione matematica sia nel "capirla".

In ogni caso non ricorreremo, in queste pagine, ad espressioni così complesse.

Le variabili

Abbiamo visto poco fa che con un personal computer è possibile realizzare le stesse funzioni di una calcolatrice tascabile con molte differenze positive: sullo schermo del calcolatore (quasi sempre rappresentato dal televisore domestico) rimangono visualizzate le ultime operazioni compiute ed è più agevole un controllo da parte dell'operatore. Quando però lo schermo viene riempito, il calcolatore effettua automaticamente il cosiddetto "scrolling", vale a dire lo scorrimento della pagina video verso l'alto.

con la conseguenza di cancellare i caratteri delle prime righe in modo da lasciare le ultime (quelle, cioè, in basso) libere di ospitare altri comandi.

Passiamo ora ad un altro argomento apparentemente slegato dal "difetto" dello scrolling. Quasi tutte le calcolatrici posseggono il tasto della "memoria" che consente di memorizzare il valore numerico presente sul visore al momento della pressione del tasto. Quante memorie possiede un personal computer?

Un numero decisamente superiore, come è facile intuire. Prima di vedere *quanti*, vediamo però *come* possono essere memorizzati alcuni valori numerici. E' però necessario rinfrescarci la memoria su ciò che si intende per "formula matematica". Essa è una relazione universale che rivela la propria utilità sostituendo ai valori letterali gli opportuni valori numerici (ma che diavolo vorrà dire?).

Ricorriamo ad un esempio. A scuola ci hanno insegnato che l'area di un rettangolo è data da:

A = b x h

in cui A rappresenta l'area che si desidera calcolare, b la base ed h l'altezza del rettangolo da studiare. Se al posto di "b" ed "h" sostituiamo valori numerici, il risultato della moltiplicazione viene chiamato Area ed indicato, per comodità, con A. Il motivo per cui gli umani si servono di lettere dell'alfabeto, invece dei simboli più propriamente numerici, risiede nel fatto che la mente umana riesce a concentrarsi più facilmente su simboli brevi anzichè su numeri. Non dobbiamo infatti dimenticare il caso di formule complesse in cui intervengono numerosi simboli. Se si utilizzassero direttamente, al loro posto, i valori numerici rappresentati, si commetterebbero facilmente degli errori. L'essere umano, invece, "lavora" inizialmente con simboli più semplici da trattare ed in seguito, dopo aver semplificato con opportuni accorgimenti la formula stessa, sostituisce i numeri ed esegue i calcoli, limitando in tal modo la possibilità di sbagliare.

Che cosa c'entra tutto questo con le variabili?

Provate a digitare:
A = 123 (R) oppure...
LET A = 123

Come risposta otterrete soltanto un READY. Tutto qui? Provate ora a cancellare lo schermo (tasto Shift insieme con Clr/home) in modo da avere solo il cursore lampeggiante, e digitate:

PRINT A (R)

ottenete in risposta: 123! Diamine, ma allora memorizzare un valore è la cosa più semplice di questo mondo! E se noi, per esempio, digitiamo

A = 456: B = 789 (R)

succede forse che...

PRINT A: PRINT B: PRINT A+B (R)

... è proprio vero: il computer tratta le lettere come se al loro posto fossero presenti i numeri prima impostati. Invece di 456+789 si può digitare PRINT A+B. A proposito, prima A valeva 123 mentre ora rappresenta il numero 456. Ne deduciamo che in "A" viene incasellato stabilmente l'ultimo valore comunicato al computer. Ecco dunque da dove viene il nome di *variabile*.

Siamo sicuri che il dato viene memorizzato stabilmente? Facciamo una prova per sincerarcene.

Spegniamo il computer, aspettiamo qualche secondo e riaccendiamolo. Provate a digitare:

PRINT A: PRINT B: PRINT C (R)

otteniamo come risposta tre zeri (0). La prova appena effettuata dimostra che:

- Quando si spegne un computer tutti i dati memorizzati vengono *irremediabilmente persi*.

- Quando un computer viene acceso tutte le variabili sono poste automaticamente al valore nullo (vedi infatti ciò che accade alle variabili "A", "B" e "C" nell'esempio appena visto).

Fermiamoci un momento prima di proseguire.

Proviamo a digitare, *carattere per carattere:*

PRINT A,B,C (R).
ed in seguito:.
PRINT A;B;C (R).

Una semplificazione notevole

dopo aver assegnato, oppure no, valori qualunque alle tre variabili (esempio: A=123: B=456 eccetera). Il risultato che si ottiene è che nel primo esempio (caso della virgola usata come separatore tra le lettere) i valori sono allineati in modo tale che la prima cifra di ciascuno di essi sia a distanza eguale dalla corrispondente *prima* cifra del valore che precede e che segue. Nel caso invece del punto e virgola notiamo che i tre valori sono trascritti l'uno dopo l'altro senza "sprecare" spazio tra un valore e l'altro. Il metodo appena descritto può esser utile sia per battere meno tasti, sia per risparmiare spazio sullo schermo, sia per allineare più valori su di uno stesso rigo.

Quante variabili può memorizzare un computer?

La risposta sembrerebbe banale: tante quante sono le lettere dell'alfabeto, vale a dire 26 (cioè A, B, C...Z). In realtà sono parecchie di più. Tutti i pers onal riconoscono infatti variabili con due lettere. Digitando ad esempio:
A = 123: AA = 456 (R)
PRINT A,AA (R)
otteniamo, appunto, 123 e 456 e ciò dimostra che la variabile "A" è considerata diversa dalla variabile "AA". Le variabili da trattare "vanno" quindi da:.

AA AB ... fino ad AZ
BA BB ... fino a BZ
... ...
ZA ZB ... fino a ZZ

per un totale di 26x26=676 variabili. Queste devono essere aggiunte alle 26 viste prima e, stavamo per dimenticarlo, perfino alle variabili del tipo:

A0 A1... A9
B0 B1... B9
.. ..
Z0 Z1 .. Z9.

Queste variabili sono cioè individuate da una lettera ed un numero.

Nota bene: non al contrario: "A0" è identificata come una variabile mentre "0A" non ha alcun significato o meglio può averne uno molto... strano cui non accenniamo nemmeno. Ricordate anche che nelle variabili con numero si avranno interpretazioni diverse a seconda se si digita A zero ("A0") oppure "AO".

In totale si dispone di:
26 + 676 + (10x26) = 962 variabili
vale a dire valori diversi che è possibile memorizzare, modificare e richiamare a piacimento. Come vedremo in seguito i valori sono di gran lunga più numerosi ed in ogni caso incomparabilmente più "comodi" da gestire in confronto ad una calcolatrice tascabile.

Una variabile può avere anche più di due lettere, ma il computer riconosce comunque solo le prime due e trascura le altre. Ad euempio per il computer "RA", "RAGGIO", e "RADAR" è sempre la stessa variabile.

Bisogna, inoltre, prestare attenzione a non adoperare, come "nomi" di variabili, gruppi di lettere dell'alfabeto che individuano particolari comandi del computer.

Se ad esempio tentate di digitare:.
COSTANTE=25 (R)
otterrete null'altro che un SYNTAX ERROR. Le prime tre lettere iniziali della variabile in oggetto (COS) sono infatti "riservate" al computer per calcolare il coseno di un angolo.

Analogamente i nomi TABELLA, FORNITURA, POSTA ed altre generano un messaggio di errore perchè iniziano con le "parole riservate" TAB, FOR, POS.

Una nuova funzione: FRE(0)

Quando accendiamo un personal computer Commodore, se proviamo a digitare:

PRINT FRE(0) (R)

otteniamo come risposta un numero intero. Questo rappresenta il numero di byte (locazioni di memoria) a nostra disposizione, in quel momento, nel calcolatore.

Nota per il Commodore 64: se con FRE(0) il valore che compare è positivo, esso indica la quantità di memoria effettivamente disponibile. Se invece risulta n egativo, è necessario battere:

PRINT 2↑16 + FRE(0) (R)

per conoscere, appunto, l'esatta quantità di memoria.

Supponiamo che il numero sia 1000. Se ora proviamo a digitare...:

A = 123 (R)

e "chiediamo" nuovamente PRINT FRE(0) otteniamo in risposta un valore inferiore a quello prima visto ed esattamente (caso di tutti i personal Commodore), inferiore di sette byte.

Facendo diverse prove ci possiamo render conto di quanto segue:

1/ Ogni nuova variabile memorizzata richiede sette celle di memoria dette "byte".

2/ La modifica del valore di una variabile precedentemente "dichiarata" non altera il numero di byte rimasti a disposizione.

3/ Il numero di byte richiesti per la memorizzazione è sempre lo stesso (sette) sia che il valore memorizzato sia di una sola cifra, come nel caso in cui sia di più cifre o addirittura contenente la virgola (che, come è noto, è un punto...).

Per renderci conto di quanto asserito facciamo la seguente prova:

- Spegnete e riaccendete il computer.
- digitare PRINT FRE(0) (R)
- A=5: PRINT FRE(0) (R)
- A=567.456: PRINT FRE(0) (R)
- A=0: B=45.87: ? FRE(0) (R)
- AA=45:A=12:B=-76:PRINT FRE(0) (R)

A che serve ciò che abbiamo appena detto?

Semplicemente per ricordare che una cosa è la potenzialità di un computer, altra cosa è l'effettiva disponibilità di tale potenzialità (ma che è, filosofia?)

Spieghiamoci meglio.

Se un calcolatore dispone, ad esempio, di soli 1000 byte di memoria non sarà possibile, e solo per motivi strettamente fisici, memorizzare un numero di variabili superiore a 142 (dato da: 1000/7) pur se in teoria il computer potrebbe farlo. Volendo sfruttare appieno le potenzialità della macchina è necessaria una memoria di almeno 962x7=6734 byte. Come però avremo modo di vedere, il numero d i variabili che è possibile teoricamente inserire in memoria è di gra lunga superiore a quello visto di 962.

Fortunatamente non capita spesso di dover tenere a mente un numero così elevato di elementi e di conseguenza anche calcolatori con poche migliaia di byte (in gergo Kilobyte o Kbyte o semplicemente K) sono sufficienti per fare "grandi" cose.

Altre variabili

Se il valore che desideriamo memorizzare è intero e compreso tra -32768 e +32767, è possibile, per memorizzarlo, ricorrere alle cosiddette variabili intere.

Il nome di queste possono, come i nomi delle variabili decimali, essere composti da una sola lettera, due lettere o da una lettera ed un numero. Devono però terminare sempre col simbolo della percentuale (%), allo scopo di differenziarle dalle variabili studiate precedentemente.

Esempi:
A%=123: AA%=567: B1%=-76
eccetera.

Sono ovviamente possibili operazioni miste facendo però attenzione a non dimenticare il carattere di due punti (:) tra le varie istruzioni.

A=123.456: A%=987: PRINT A*A% (R)

Viene generato un messaggio di errore nei casi in cui non si rispettano le caratteristiche tipiche delle variabili intere.

Esempi:
A%=-40000: A%=1000*100:
A%=125.2

Nel primo e nel secondo caso, infatti, i valori superano l'intervallo lecito, mentre il terzo caso tenta di "trattare" una variabile decimale. Pur se nei primi due casi viene, comunque, visualizzato un messaggio di errore (ILLEGAL QUANTITY ERROR), nel terzo caso non viene emesso alcun messaggio. Se però digitiamo:

A%=125.25:PRINT A%

otteniamo in risposta 125: E' stata cioè memorizzata la sola parte intera. E' da notare che, anche in questo caso, è possibile memorizzare, oltre alle variabli decimali viste prima, quasi 962 variabili intere a patto, ovviamente, che vi sia memoria RAM disponibile sul personal computer adoperato.

Ancora variabili: vettori e matrici

Se la possibilità di memorizzare le centinaia di valori prima visti appare modesta oppure scomoda da gestire, vi potete servire delle variabili cosiddette "ad indice".

Questo modo di indicarle deriva proprio dal fatto che ci si può riferire ad un indice (racchiuso tra parentesi) che le distingue l'una dall'altra. Ricorriamo ad alcuni esempi prima di approfondire l'argomento:

A(1)=12: A(2)=34.54: A(3)=-45.76:.... A(9)=567 (R) eccetera.

Naturalmente sono lecite definizioni del genere:

AA%(7)=-345: AA(6)=-45*AA(2) ecc.

E' inutile dire che è possibile servirsi di due lettere qualunque per definire ciascun vettore e che il valore tra parentesi (indice) può, in genere, variare tra 0 e 65535.

Come versatilità non c'è male!

Si ricorda che se l'indice che intendiamo utilizzare è superiore a 10 bisogna, *prima* di usare le variabili, definire la dimensione massima del vettore che le conterrà. In altre parole supponiamo di voler memorizzare 23 valori servendoci non di variabili con nome diverso l'uno dall'altro, ma ricorrendo ad una variabile che ha sempre lo stesso nome ma indici di riferimento ovviamente diversi l'uno dall'altro. Se proviamo a digitare...:

A(0)=344 (R)
A(1)=567 (R)
....
A(10)=567 (R)

.... queste cifre vengono accettate. Non appena proviamo a memorizzare il dodicesimo elemento (compreso lo "zeresimo"), come ad esempio...:

A(11)=654 (R)

...otteniamo in risposta il messaggio di errore:

?BAD SUBSCRIPT ERROR

Avremmo dovuto dimensionare fin all'inizio il vettore (=tabella) che in seguito avremmo utilizzato per memorizzare i diversi valori. La sintassi per tale operazione è DIM. Se ad esempio desideriamo definire due vettori, l'uno decimale di 23 elementi col nome X1 e l'altro intero di 234 elementi col nome FE% dobbiamo digitare:

DIM X1(22): DIM FE%(233) (R)

Dopo tale operazione è possibile memorizzare...

FE%(43)=-7654: X1(21)=56.89

... oppure leggere...

PRINT FE%(143), X1(13) eccetera.

E' altresì possibile dimensionare matrici pluridimensionali. Queste sono delle vere e proprie tabelle in cui memorizzare, in ciascuna casella da cui sono formate, altrettanti valori numerici. Se per esempio digitiamo:

DIM A(20,30)

non facciamo altro che riservare spazio in memoria per un numero di 20*30=600 valori decimali. Sono pertanto possibili memorizzazioni del tipo:

A(12,5)=876: A(7,26)=987 e simili.

In parole molto semplici, con l'esempio accennato si crea in memoria lo spazio per una tabella numerica formata da 21 righe e da 31 colonne. Se si pensa che sono possibili definizioni del tipo:

DIM A(32,54,13)

è facile immaginare le enormi possibilità offerte da un personal computer. Si ricorda, però, che è bene fare i conti con la memoria disponibile. Il computer, infatti, può memorizzare a patto che... abbia memoria RAM sufficiente per compiere l'operazione! Ricordando che, in genere, una variabile decimale ad indice occupa in media cinque byte, è bene fare un calcolo prima di dimensionare un valore eccessivo. In caso contrario si ottiene in risposta un messaggio del tipo:
?OUT OF MEMORY ERROR
vale a dire: non c'è memoria sufficiente per compiere l'operazione.

Ancora variabili, stavolta alfanumeriche

Tutti sanno che un computer può "trattare" simboli anche diversi da quelli tipici della matematica e lo dimostra il fatto che gli errori vengono visualizzati, in genere, mediante veri e propri messaggi.

Anche noi possiamo memorizzare parole, e addirittura intere frasi e messaggi vari con limitazioni decisamente minime. La sintassi corretta per eseguire l'operazione consiste nel far seguire le lettere del nome della variabile dal segno del dollaro ($) e nell'inserire tra virgolette o apici (") il messaggio da memorizzare. Esempio:
A$="MESSAGGIO":
FR$="QUESTA E' UNA FRASE" (R)

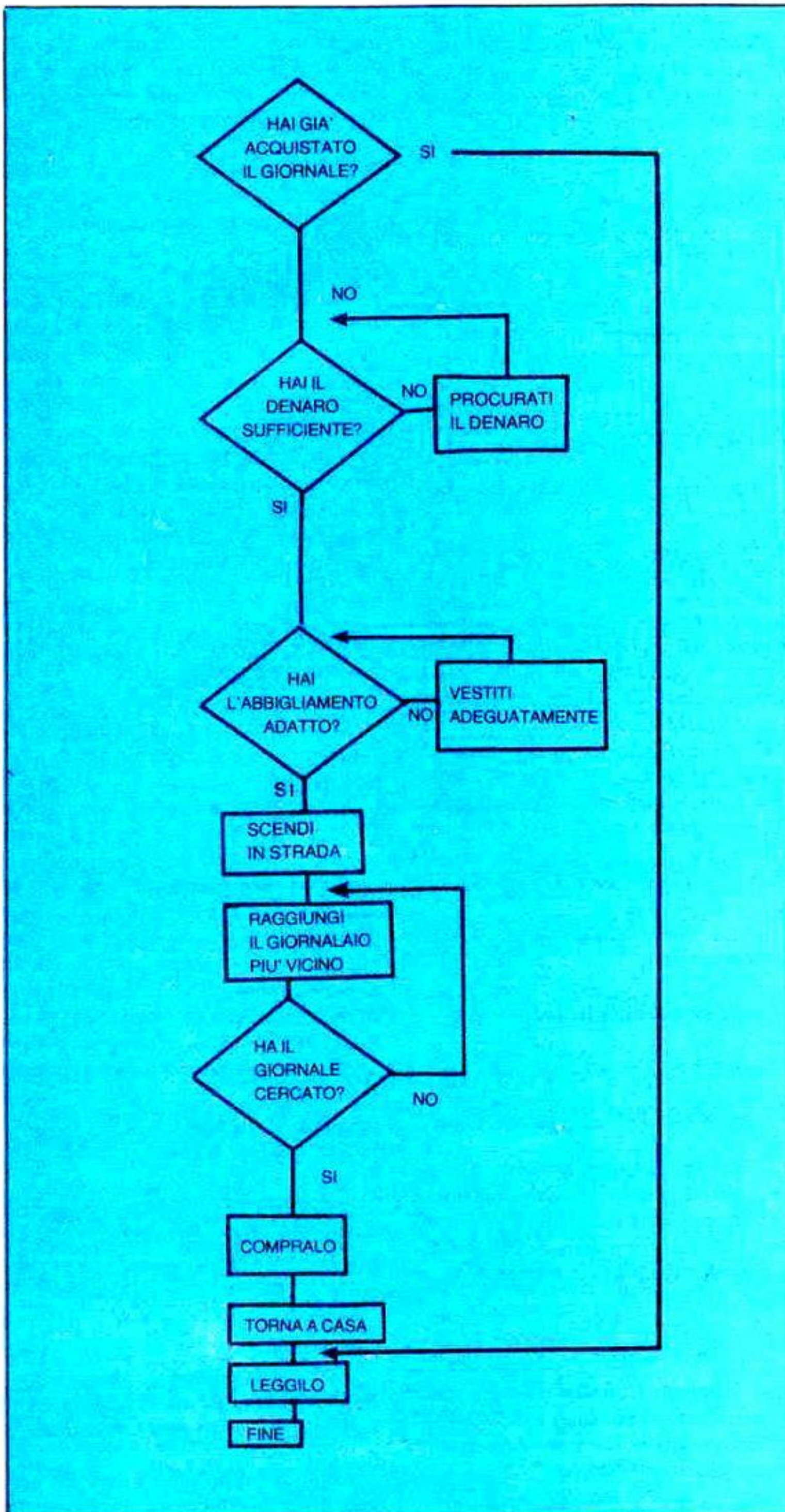
In seguito saranno ovviamente possibili operazioni del tipo....:

PRINT A$,FR$.... ed operazioni miste...

PRINT A,A%,AA$,A1.

Segnalazioni di errore nel trattamento delle variabili

Come abbiamo visto, esiste una notevole capacità di memorizzare un gran nu-

mero di variabili. E' d'altronde doveroso prestare la massima attenzione affinchè la memorizzazione avvenga senza inconvenienti di sorta. Riportiamo pertanto un elenco di messaggi di errore che il computer visualizza nel caso venga digitato un comando inesatto come nel caso di impossibilità di eseguire l'operazione richiesta.

Negli esempi che seguono il simbolo (R) significa, come al solito, "Premere il tasto di ritorno carrello (Return)".

- *?SYNTAX ERROR*

La sintassi richiesta non è stata rispettata. Casi tipici sono rappresentati dall'omissione di parentesi, segni di eguale, apici, doppi punti. Esempi:
A=(R) Manca il valore da assegnare ad A
A+B (R) Manca il comando PRINT
? (A+B)*45) (R) Il numero di parentesi aperte è diverso da quello di parentesi chiuse.
PRINT A PRINT B (R) Manca il carattere dei due punti tra le due istruzioni
PRINT A,:B, (R) C'è un carattere di doppio punto "in più".

- *?DIVISION BY ZERO ERROR*

Capita quando si tenta di dividere un qualsiasi valore per zero, operazione notoriamente impossibile. Esempi:
A=0: PRINT 100/A (R)

E' invece possibile:

A=0:PRINT A/100 (R)

- *?ILLEGAL QUANTITY ERROR*

Capita quando si richiede di trattare un valore inaccettabile. Esempi:
A%=100000 (R) Con variabili intere il valore memorizzabile deve infatti esser compreso nell'intervallo -32768 e +32767
A=-4: PRINT SQR(A) (R) Ricordiamo che SQR(X) calcola la radice quadrata di un numero. Questo deve, come è noto, essere maggiore o eguale a zero.
A=-3.76: PRINT LOG(A) (R) Vedi osservazione precedente.
DIM A(32): PRINT A(45) (R) Non è possibile riferirsi ad un indice superiore al massimo dichiarato.

- *?OUT OF MEMORY ERROR*

Capita quando si cerca di memorizzare troppe variabili o matrici con un numero eccessivo di elementi. E' intuitivo che il

messaggio di errore viene visualizzato nel caso si superi la quantità di memoria disponibile.

Consideriamo infatti il seguente esempio:

DIM A(1000) (R)

Questo comando porta al messaggio di errore in oggetto nel caso di un computer con scarsa memoria (es. Vic 20 inespanso), mentre viene accettato nel caso di computer con maggiore capacità (Commodore 64, Vic 20 con scheda di espansione di almeno 8K, Plus-4, eccetera).

- *?OVERFLOW ERROR*

Capita quando si cerca di "trattare" un valore troppo grande per la capacità di calcolo del computer. Esempio:

PRINT 10000↑10000 (R)

Ne approfittiamo per ricordare che il simbolo freccia in alto (↑) viene utilizzato per compiere l'elevazione a potenza. Se ad esempio si desidara calcolare il cubo della variabile "A" si deve digitare:
PRINT A↑3 (R)

- *?REDIM'D ARRAY ERROR*

Capita quando si tenta di dimensionare nuovamente un vettore già dimensionatoto in precedenza. Esempio:
DIM A(23): DIM A(12) (R)

- *?TYPE MISMATCH ERROR*

Capita quando si cerca di trattare come variabile numerica una stringa o viceversa. Esempio:
A="PROVA" (R) Manca il simbolo del dollaro
B$="123": A=!45+B$ (R) Non è possibile sommare numeri a stringhe anche se queste rappresentano caratteri numerici.
A$=NOME (R) Mancano gli apici di apertura e chiusura ("NOME").

Esercizi di riepilogo

Cercate di rintracciare, tra le "linee" che seguono, quelle che contengono errori, e, tra queste, determinate il messaggio di errore che il computer emetterebbe. Nel caso i comandi vi sembrino corretti, individuate la risposta che fornirebbe il calcolatore. Dovete immaginare che il computer sia stato spento ed in seguito riacceso prima di eseguire, l'uno dopo l'altro, gli esercizi proposti.

E' ovvio che le scritte 1/ 2/... 9/ non devono essere trascritte.

1/ PRINT A(7)*A(2) (R)
2/ PRINT A(7)/A(2) (R)
3/ DIM A(15) (R)
4/ A=A*34+B:PRINT A (R)
5/ CS$="VOCABOLO" PRINT CS$ (R)
6/ PRINT CS$
7/ CS$="VOCABOLO": PRINT CS$ (R)
8/ CS$=CS!+CS! (R)
9/ PRINT CS$,A,A(3) (R)

Risposte

1/ La prima riga è corretta. Il risultato fornisce zero. Infatti non appena si accende il computer tutte le variabili valgono zero. Si noti soprattutto che il semplice fatto di aver "nominato" variabili con indice, genera un dimensionamento implicito. Ce ne accorgeremo digitando il comando della terza riga.

2/ Deriva un DIVISION BY ZERO ERROR: Non si può dividere un numero (A(7)) per zero (A(2)).

3/ Viene visualizzato il REDIM'D ARRAY ERROR dato che precedentemente (rigo uno) è stato già definito, anche se implicitamente, il vettore dal nome A.

4/ Il risultato è zero. Anche in questo caso "A" e "B" valgono zero non appena si accende il computer.

5/ Viene visualizzato un SYNTAX ERROR. Manca infatti il segno dei due punti (:) tra i due comandi (prima di PRINT). Si noti che in questa fase, nonostante l'omissione del carattere si manifesti *dopo* un comando corretto (CS$="VOCABOLO"), questo non viene memorizzato. Nel caso, infatti, in cui su di una stessa riga vengano digitati più comandi, sono accettati solo quelli sintatticamente corretti fino al carattere di doppio punto (:). Nel caso presente non vi sono comandi corretti prima del... primo doppio punto (che addirittura manca).

6/ La battitura di questa riga, che fornisce in risposta un rigo vuoto sullo schermo, dimostra che la riga precedente (5/) non è stata accettata nemmeno in parte. Per il computer la stringa CS$ è "vuota", non contiene, cioè alcun carattere.

7/ Questo è il modo corretto di digitare la riga 5. Viene ovviamente visualizzata la parola VOCABOLO.

8/ Viene eseguita la... somma di due parole. Vale a dire che mentre prima la stringa CS$ conteneva la parola vocabo-

lo, dopo l'operazione di somma conterrà il risultato dell'operazione presente a *destra* del segno eguale, vale a dire: VOCABOLOVOCABOLO.

Non è possibile un'operazione di differenza, moltiplicazione o divisione tra stringhe che, come è intuitivo, non avrebbero sanso.

9/ La visualizzazione di CS$ viene seguita da due zeri, valori posseduti dalle variabili A e A(3). Si fa notare che nella fase 8 non viene visualizzata la stringa CS$ per il semplice motivo che... non è stato richiesto. Si deduce che il computer esegue *soltanto* quei comandi che vengono impartiti. Se si desidera visualizzare qualcosa è indispensabile ricorrere al comando PRINT. In ogni caso il risultato delle varie elaborazioni viene mantenuto in memoria fino a che non si spegne la macchina o vengono modificate le variabili stesse mediante opportune istruzioni.

C'è un'istruzione che cancella tutte le variabili, di qualunque tipo esse siano. Questa è "CLR" (abbreviazione di Clear=cancella).

Attenzione quindi ad usarla!

La programmazione

La programmazione non è, come comunemente si crede, un concetto nuovo nè richiede, per la comprensione, altre esperienze al di fuori di quelle quotidiane. Ricorreremo subito ad un esempio banalissimo per dimostrare quanto asserito.

Supponiamo di voler andare in edicola ad acquistare un giornale. Per raggiungere lo scopo prefissato (acquisto di un giornale) noi eseguiamo *sempre*, magari senza rendercene conto, una serie di operazioni. Volendo descrivere il nostro comportamento in modo sbrigativo, possiamo riassumere le azioni da compiere nel modo seguente:

1/ scendiamo in strada;
2/ ci rechiamo all'edicola;
3/ acquistiamo il giornale;
4/ torniamo a casa.

Forse non ci crederete, ma questo appena descritto è un programma! Esso è stato, per pura questione di comodità e chiarezza, suddiviso in quattro punti "fondamentali" ma potevano anche essere in numero inferiore:

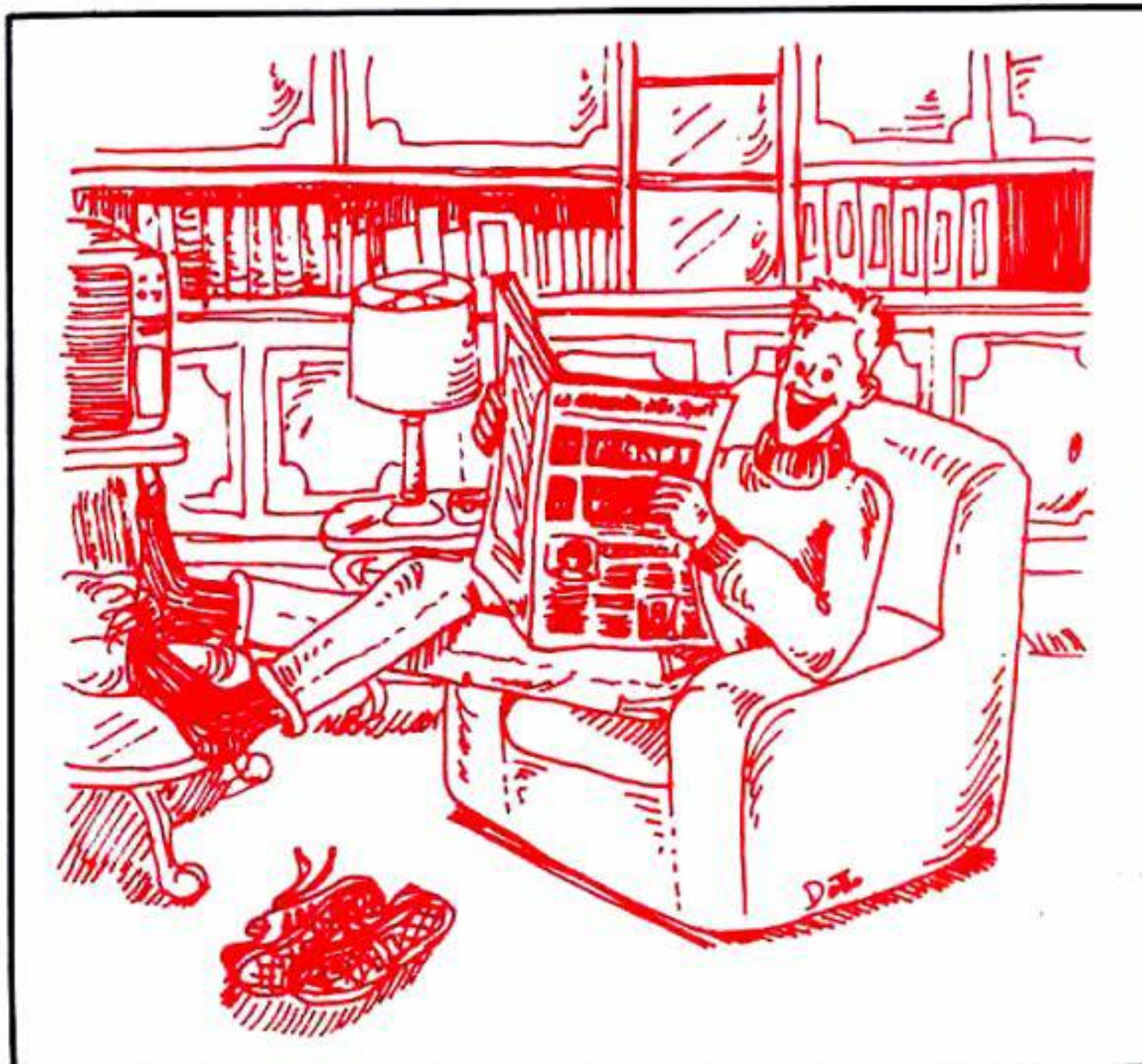
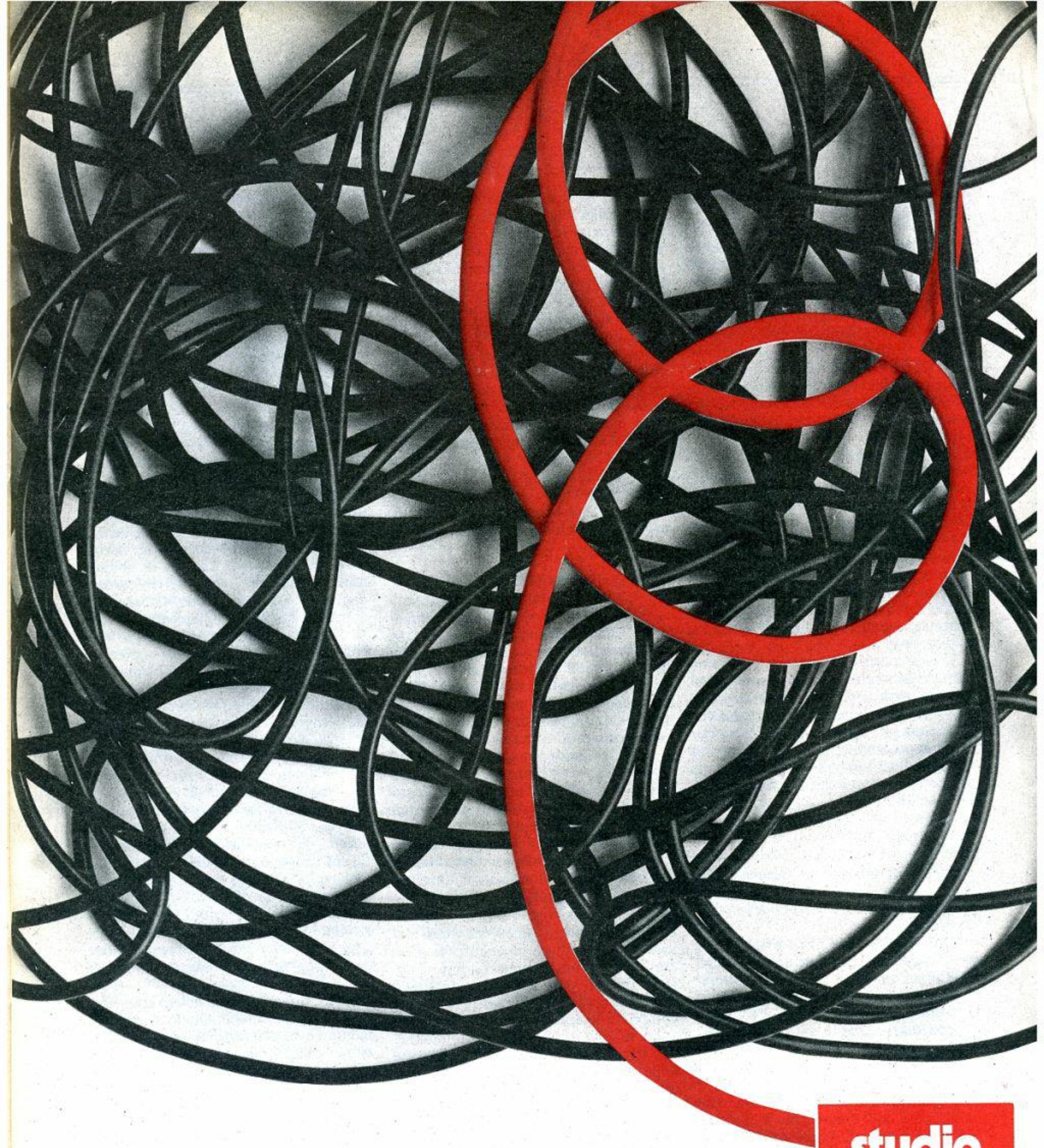
1/ ci rechiamo dal giornalaio per acquistare il giornale;
2/ torniamo a casa a leggerlo.

A voler essere un po' pignoli i punti prima visti dovrebbero essere ben più numerosi.

Scommettiamo che, nell'azione abituale di acquisto, eseguite tutte le operazioni che tra breve verranno descritte? Osservate attentamente. Vi sfidiamo a negare anche una sola delle operazioni che compite quando vi "programmate" per l'acquisto di un giornale:

1/ Verificate se il giornale desiderato non l'avete già acquistato.
2/ Se lo avete già comprato saltate alla fase N.16, altrimenti continuate con la N.3.
3/ Controllate di avere in tasca il denaro sufficiente per l'acquisto.
4/ Se la verifica della fase N.3 è negativa vi procurate il denaro. Se non riuscite a procurarvelo saltate alla fase N.17.
5/ Controllate che il vostro abbigliamento e/o aspetto personale siano adatti... alla missione che state per compiere.
6/ Nel caso in cui siete in vestaglia e pantofole provvedete a cambiarvi a seconda della stagione e delle... condizioni atmosferiche (se piove prendete l'ombrello).
7/ Se c'è qualcuno in casa avvertite che state per uscire.
8/ Usate le scale o l'ascensore a seconda dei casi (fermo per manutenzione, mancanza di ascensore ecc).
9/ Decidete quale giornalaio raggiungere (probabilmente il più vicino).
10/ Se è necessario attraversare la strada mettete in atto le più elementari regole di prudenza nell'attraversarla.
11/ Raggiunto il luogo chiedete ciò che desiderate.
12/ Se la pubblicazione è esaurita individuate l'altra edicola più vicina. Saltate alla fase 9.
13/ Pagate la cifra richiesta ed intascate l'eventuale resto.
14/ Riprendete la strada di casa.
15/ Entrate nel vostro appartamento.
16/ Vi mettete a leggere (finalmente) il giornale.
17/ Fine di tutte le operazioni.

Se avete avuto la pazienza di seguirci fin qui, vi sarete sicuramente accorti che è possibile essere ancora più pignoli di quanto non lo siamo stati noi stessi. Ad esempio tra la fase 14 e 15 sarebbe opportuno inserire una fase (N.14a) del tut-

studio
d

STUDIO D
PER NON SMARRIRE MAI IL FILO DEL DISCORSO.
STUDIO D
EMITTENTI RADIOTELEVISIVE INDIPENDENTI CHE SI FANNO SENTIRE.

CONCESSIONARI MEZZI
RADIOTELEVISIVI

STUDIO D
Via Rossini 5 - 20122 MILANO
Tel. (02) 799.592-782.503

to simile alla 10. La prudenza nell'attraversare la strada è sempre auspicabile! Altre fasi possono essere introdotte a piacivento nel considerare casi particolari. Ad esempio se piove a dirotto è probabile che vi fermiate alla fase 4 rinunciando all'acquisto oppure... incaricando un vostro familiare di scendere in strada!

In effetti si dovrebbero prevedere altri casi particolari come l'incontro casuale di un conoscente che potrebbe distrarvi facendovi rinunciare alla missione da compiere.

In definitiva, quando decidiamo di compiere qualche azione, qualunque essa sia, ci organizziamo, (ripetiamo: il più delle volte senza rendercene conto) secondo un "programma", vale a dire una serie di azioni logiche, da eseguire in successione l'una dopo l'altra.

In ogni caso ci sarà un'azione da compiere per prima ed una che rappresenta l'ultima della lista. Nel caso l'azione da compiere risulti diversa a seconda del verificarsi o meno di un evento (caso delle fasi N.2/ 4/ 7/), ci troveremo di fronte ad una scelta da effettuare tra le varia possibili, l'una delle quali in genere diversa dall'altra.

Il grafico che deriva dallo studio delle azioni da compiere prende il nome di *diagramma di flusso* appunto perchè rappresenta il fluire "logico" delle azioni da compiere nel corso dello svolgimento del compito da espletare. Quelli relativi ai due schemi appena visti sono evidenziati nelle figure.

In queste ogni azione è incasellata in una forma geometrica unificata di cui descriviamo l'uso qui di seguito:

- *Rettangolo*. Indica un'azione da compiere senza... discussioni.
- *Rombo*. Implica una scelta da compiere. La direzione del flusso prende una via oppure un'altra a seconda delle condizioni in cui si trova l'evento da considerare. Le deviazioni possono essere molteplici e non solo due come comunemente si crede. Avremo occasione di verificarlo in seguito.
- *Ovale*. Indica l'inizio oppure la fine delle operazioni.

Una breve precisazione

Abbiamo detto che un programma non è altro che un elenco di azioni da svolgere in successione. Per distinguere la precedente dalla successiva, si ricorre alla numerazione delle azioni stesse come, appunto, abbiamo fatto in precedenza. Anche il calcolatore va programmato allo stesso modo ma, negli esempi che seguono, tale numerazione sarà un pò bizzarra. Infatti i numeri che individuano le azioni (o gruppi di azioni da compiere), non saranno consecutivi, nè si partirà dal numero 1. Il primo schema visto, per esempio, lo scriveremo come segue:

100/ Scendiamo in strada.
110/ Ci rechiamo all'edicola.
245/ Acquistiamo il giornale.
280/ Torniamo a casa.

Il perchè di tale insolita numerazione risiede nel fatto che ci riserviamo la possibilità di inserire, nel seguito, alcune fasi che possiamo aver dimenticato di inserire. Se riteniamo, ad esempio, che è indispensabile controllare anzitutto il possesso del denaro, potremo agevolmente inserire una riga di azioni numerata con un numero inferiore a 100 (prima riga del caso appena visto). Questa può essere la seguente:

90/ Controllo del possesso del denaro necessario all'acquisto.

Se fossimo partiti direttamente col numero uno, tale inserimento non sarebbe stato possibile se non cambiando la numerazione a tutte le fasi con notevole perdita di tempo. Analogamente altre linee di azioni possono essere inserite tra altre linee, purchè la numerazione di queste lo consenta.

In ogni caso, anche ricorrendo all'insolito elenco, viene sempre rispettato l'ordine logico. L'importante è, infatti, fare dapprima l'azione che possiede il numero più basso; subito dopo quella col numero immediatamente maggiore e di seguito tutte le altre fino all'ultima.

Sul prossimo numero di Commodore Computer Club, in cui concluderemo la "chiacchierata" coi principianti, trasferiremo su calcolatore i concetti appresi e scriveremo, finalmente, il nostro primo programma.

Nel frattempo, se avete in casa un Commodore, digitate e fate girare i programmi pubblicati su questo numero. Anche se non capite granchè del loro funzionamento, vi accorgerete che utilizzare un computer è molto più semplice di quanto non pensiate.

La virgola è mobile

Claudio Baiocchi

> *In inglese si chiama floating point. Per noi, è la virgola mobile. Serve per memorizzare le variabili. Ecco come usarla.*

La memorizzazione delle variabili reali nei computer Commodore avviene utilizzando la virgola mobile (o, dall'inglese, floating point).

In questo articolo descriviamo tale tecnica, la cui comprensione può in particolare permettere di prevedere e prevenire i problemi legati alla precisione (o meglio all'imprecisione) che ne deriva (vedi il riquadro).

Per quanto riguarda gli altri tipi di variabili, i riquadri a corredo forniscono qualche complemento all'articolo "Cinque Utility in Basic" (CCC N. 15 programma 2) e richiamano alcuni oggetti già illustrati nell'articolo "un po' d'ordine tra i Bit" (CCC N. 10).

Se digitiamo:

X = 1000000000 : PRINT X

il Commodore passa automaticamente alla cosiddetta notazione scientifica, scrivendo 1 E +09.

In genere in questo tipo di scritture il simbolo E sta per "moltiplicato per 10 alla"; il numero che segue la E è detto esponente, e deve essere un numero intero; al contrario il numero che precede la E può avere una parte decimale (eventualmente periodica, o anche peggio...).

Naturalmente uno stesso numero può avere varie scritture, ad esempio:

234 E-2 = 23.4 E-1 = 2.34 E 0 = 0.234 E 1 = 0.0234 E 2 =...

ma tra tutte queste ne esiste una sola che rispetta i due requisiti seguenti:

(i) la scrittura inizia con zero virgola (0.) ovvero con meno zero virgola (-0.)

(ii) la prima cifra dopo la virgola non è 0;

Quando tali condizioni sono rispettate si dice che il numero è stato scritto in forma normalizzata (si osservi che lo zero non possiede forma normalizzata).

Nel Basic Commodore le variabili reali sono rappresentate in forma normalizzata: ma naturalmente in base 2, non in base 10! Ciò significa che le cifre dopo la virgola, anzichè i decimi, centesimi etc., rappresentano i mezzi, i quarti, gli ottavi e così via. Inoltre il simbolo E sta ora per "moltiplicato per 2 alla".

Per capire come un numero verrà memorizzato bisognerà perciò effettuare due trasformazioni: da decimale a binario, poi da binario a binario normalizzato. Ad esempio:

-5 -101 -0.101E3

3.75 11.11 0.1111 E 2

4/3=1. (3) 1. (01) 0.(10) E 1 (periodico)

0.1 0.0(0011) 0.(1100) E3 (periodico)

Il fatto che il decimale 0.1 risulti periodico in numerazione binaria è causa di molti guai... (vai ancora a vedere il riquadro).

Una volta rappresentato il numero in binario normalizzato, la tecnica del Basic Commodore per memorizzarlo (cioè per costruire i numeri V1,...,V5: notazioni del riquadro a pag. consta di 3 fasi:

- L'esponente deve essere compreso tra -127 e +127. Se ciò è vero, si pone V1 = esponente +128 (è un intero tra 1 e 255, che può quindi essere "pokato" nella cella voluta).

Quando l'esponente risulta > 127 viene emesso il messaggio "?OVERFLOW". Se infine l'esponente è < -127 si pone semplicemente V1 = 0; e quando V1 = 0 (indipendentemente dai valori as-

sunti da V2,...,V5) il valore della variabile in questione è assunto essere 0 per convenzione.

Si osservi che, grazie a questa convenzione, risulta possibile trattare anche il numero 0 (che non ha forma normalizzata); e risultano velocissimi test del tipo

IF X = 0 THEN

e

IF X < > 0 THEN

richiedendo la sola lettura di V1.

- I restanti valori V2,V3,V4,V5 sono visti come un unico blocco di 32 bit. Nel secondo di essi si immette la seconda cifra dopo la virgola; nel terzo la terza e così via. Se non ci sono 32 cifre dopo la virgola, si aggiungono zeri. Se la cifra 33 fosse un 1 si incrementa la 32, con eventuali riporti sulla 31 etc.

Non abbiamo ancora provveduto a memorizzare: il segno, la parte prima della virgola, e la prima cifra dopo la virgola. Ed è qui che si rivela la potenza della scrittura normalizzata.

La parte intera è inutile memorizzarla: la regola (i) ci dice che essa è comunque 0.

La prima cifra dopo la virgola è inutile memorizzarla: la regola (ii) ci dice che essa non è 0; ma in binario ciò significa che essa è un 1.

Il segno viene memorizzato nel primo dei 32 bit a disposizione, con convenzione analoga a quella usata per i numeri interi: V1 AND 128 varrà 0 per i positivi, e 128 per i negativi.

Naturalmente, al momento dei conti, una volta valutato il segno del numero, occorrerà ripristinare l'1 in prima posizione (si dice che tale bit è mascherato dal segno). Complicato? Forse, ma tanto a fare i conti ci pensa il Computer...; il tutto è simulato da Basic nel programma di pag.)la ricostruzione del numero può essere affetta da approssimazioni, ma la scrittura binaria normalizzata è esatta.

Il programma è puramente didattico: le linee 330, 340 possono essere sostituite da INPUT A: GOTO 150 ma così facendo si perde la possibilità di assegnare ad A il risultato di un calcolo. Si faccia attenzione al fatto che la variabile A deve essere la prima incontrata dal programma.

Al RUN, la variabile A vale 0.1. Se ne è già vista la scrittura binaria normalizzata: 0.(1100) E-3. La cella esponente contiene perciò -3+128=125. Le celle mantissa dovrebbero essere tutte 11001100 binario = 128+64+8+4 = 204, ma nel primo byte occorre fare un AND 127 (segno +) e nel quarto occorre fare un +1 (il primo bit trascurato è un 1). I cinque byte mostrati dal programma sono perciò 125,76,204,204,205.

Il programma chiede ora se vogliamo cambiare A. Rispondiamo "SI" ed assegniamo ad A il valore -5. Abbiamo visto che in binario normalizzato -5 si scrive -0.101 E3. Aggiunti gli zeri finali, e lasciato a 1 il bit segno si avrà V1=3+128=131; V2=binario 10100000 = 128+32 = 160 V3,V4,V5 sono nulli ed il programma mostra appunto i valori 131,160,0,0,0:

Rispondiamo ora "NO" alla richiesta di cambiare A, e "SI" alla richiesta di cambiare V1,...,V5 e proviamo a "forzare" in A il valore -4/3 agendo sui V1,...,V5. Sappiamo che -4/3 si scrive -0.(10) E1 nella cella esponente dobbiamo perciò inserire 1+128=129 mentre i byte

I vettori

Il puntatore (47,48) ((44,45) sul PET) oltre che alla fine delle variabili punta all'inzio degli array. La fine degli array è puntata da (49,50) (da (46,47) sul PET).

Ai due byte che indicano nome e tipo dell'array (codificati come per le variabili qui il quarto tipo non esiste) seguono due byte che esprimono (nel solito modo: parte bassa, poi parte alta) il numero di celle riservate alla memorizzazione dell'array. Ovviamente lo spazio occupato dipende da come è stato dimensionato l'array.

Segue poi un byte che esprime la dimensione dell'array, cioè il numero di indici: 1 per i vettori, 2 per le matrici.

Per ogni indice (a partire dall'ultimo) una coppia di byte che fornisce il numero di valori che l'indice stesso può assumere (1 in più del numero usato per dimensionare: s ricordi che ogni indice può assumere anche il valore 0).

Al termine di queste informazioni, inizia lo spazio in cui i singoli elementi dell'array sono memorizzati. Negli array a più indici vengono per primi gli elementi che corrispondono all'ultimo indice nullo. Tra questi i primi sono quelli con penultimo indice nullo. In particolare, le matrici sono memorizzate "per colonne".

Nella memorizzazione di un array reale, ogni elemento richiede ancora 5 celle per i V1,...,V5 ma per un array di interi gli elementi occupano solo due byte (V1 e V2), e per gli array di stringhe solo 3 (V1,V2,V3). In particolare, dovendo lavorare con grandi quantità di numeri interi (compresi tra 2 15 e 2 15-1) conviene usare un array di interi, anziché di reali, per via del notevole risparmio di memoria che ne consegue.

Al contrario le variabili intere occuppano 7 byte, come quelle reali; ci si potrebbe però aspettare che un'operazione come X% = Y% + Z% venga svolta più rapidamente della analoga X = Y + Z.

In realtà il Basic Commodore non dispone di apposite subroutine per il trattamento delle operazioni tra interi: esegue dapprima una conversione in virgola mobile, poi l'operazione, e infine riconverte il risultato in intero...

In definitiva lavorare con le variabili intere si riduce solo ad una perdita di tempo!

mantissa sono tutti binario 10101010 = 128+32+8+2=170. Il primo bit del primo byte non va modificato (segno -) mentre all'ultimo byte va sommato 1 (primo bit soppresso =1).

Alla richiesta di V1,...,V5 forniremo perciò 129,170,170,170,171; e vedremo apparire A=-1.33333333.

Come ultimo esperimento proviamo a forzare in A il più grande numero che il Commodore può gestire: V1=255, V2=127 (segno +), V3=255, V4=255, V5=255: l'esatto valore di A è dato allora dalla formula:
2 . 127*(1/2+1/4+1/8+....+1/2↑32) = 2 127*(1-2↑-32) = 2↑127-2↑95.

Inserendo i valori 255,127,255,255,255, come V1,...,V5 il Commodore fornirà A = 1.70141183 E+38 ma nella ricostruzione di tale valore le approssimazioni effettuate nei calcoli intermedi generano un overflow.

Veniamo ora alla linea 1000 ed alla misteriosa linea 9980. La subroutine 10000 converte un numero Z (intero, tra 0 e 255) nella scrittura binaria a 8 bit: con RUN 1000 si può utilizzare (correttamente) tale subroutine.

La linea 9980 è un'alternativa (a priori lecita, ma pericolosa) alla linea 1000: con RUN 9980 oppure GOTO 9980 si otterrà la conversione, seguita ovviamente da un "?RETURN WITHOUT GOSUB". Però, eseguendo GOSUB 9980 tutto dovrebbe funzionare...

In realtà, esguendo GOSUB9980 (senza abbreviare il GOSUB e senza apazio tra esso e il 9980) tutto funziona bene se, alla richiesta input, si fornisce un numero di due o tre cifre. Se invece si fornisce un numero di una cifra appare un incredibile "?SINTAX ERROR IN 48"
Ne riparleremo...

Le variabili in memoria

I tipi di variabili sono quattro:
Tipo reale: N1 < 128. L'interpretazione di V1,...,V5 è data nell'articolo.
Tipo intero: N1 > 127. La quantità V1*256+V2, diminuita di 2 16 se V1 > 128, è il valore della variabile.

I valori V3, V4, V5 non sono utilizzati.

Tipo stringa: N1 < 128, N2 > 127. La stringa ha la lunghezza V1; se V1 > o essa è memorizzata in V1 celle a partire da quella numero V2+256*V3 V4 e V5 non sono utilizzati.

Tipo funzione: N1 > 127, N2 < 128. Se ad esempio è stato eseguito il comando DEF FN (X) = 1 + X, la zona BASIC in cui tale definizione e tokenizzata si trova a V1+256*V2 (in tale cella è tokenizzata la prima istruzione a destra del segno "eguale" "="); V5 è il token della prima istruzione (nell'esempio: V5 = 48 , token della cifra 1); infine V3+256*V4 è la cella dove si trova il dato V1 della variabile utilizzata (nell'esempio: il dato V1 della variabile X).

Riportiamo, qui di seguito, un breve compendio dell'articolo pubblicato sul n. 10 di C.C.C. "un po' d'ordine tra i Bit".

A partire dalla cella "puntata" da (45,46), cioè dalla locazione di memoria PEEK (45) +256*PEEK (46), e fino alla cella puntata da (47,48), ogni blocco di 7 byte è riservato alla memorizzazione di una variabile (nel PET i puntatori corrispondenti sono (42,43) e (44,45).

Indicheremo con:
N1,N2,V1,V2,V3,V4,V5 i valori (ottenibili tramite PEEK) contenuti in un blocco.

N1 e N2 contengono informazioni sul nome e sul tipo della variabile; V1,...,V5 servono a ricostruirne il valore. Precisamente:
N1 AND 127 è il codice ASCII della prima lettera del nome;
N2 AND 127 è il codice ASCII della seconda lettera (oppure è 0 per le variabili con nome corto, quali X,A%,N$)

Strano, ma vero

Il fatto che, fra le tante, l'assegnazione:
X = SQR (9) - 3
fornisca un valore < > 0 per X ovviamente da imputare alla scarsa precisione della routine SQR del BASIC Commodore.

Al contrario, il fatto che ponendo:
Z = 5 : Y = SQR (Z) : X = Y*Y - Z
si ottenga X < > 0 è inevitabile: nessun computer può esguire esattamente l'assegnazione: Y = SQR (5)!

Talvolta può accadere che due approssimazioni si compensino: ad esempio sostituendo nel comando precedente il 5 con un 2, si ottiene X = 0. Ma è importante che il programmatore sappia "fiutare" il pericolo...e non sempre la cosa è facile.

Ad esempio il numero 0.1 è un numero "innocente" in base 10, ma in base 2 risulta periodico, quindi il Commodore deve fare un'approssimazione per eseguire LET X = 0.1.

E' per tale motivo che un ciclo come:
FOR X = 0.1 TO 1 STEP .1 : ..... : NEXT
viene eseguito nove volte anziché dieci.

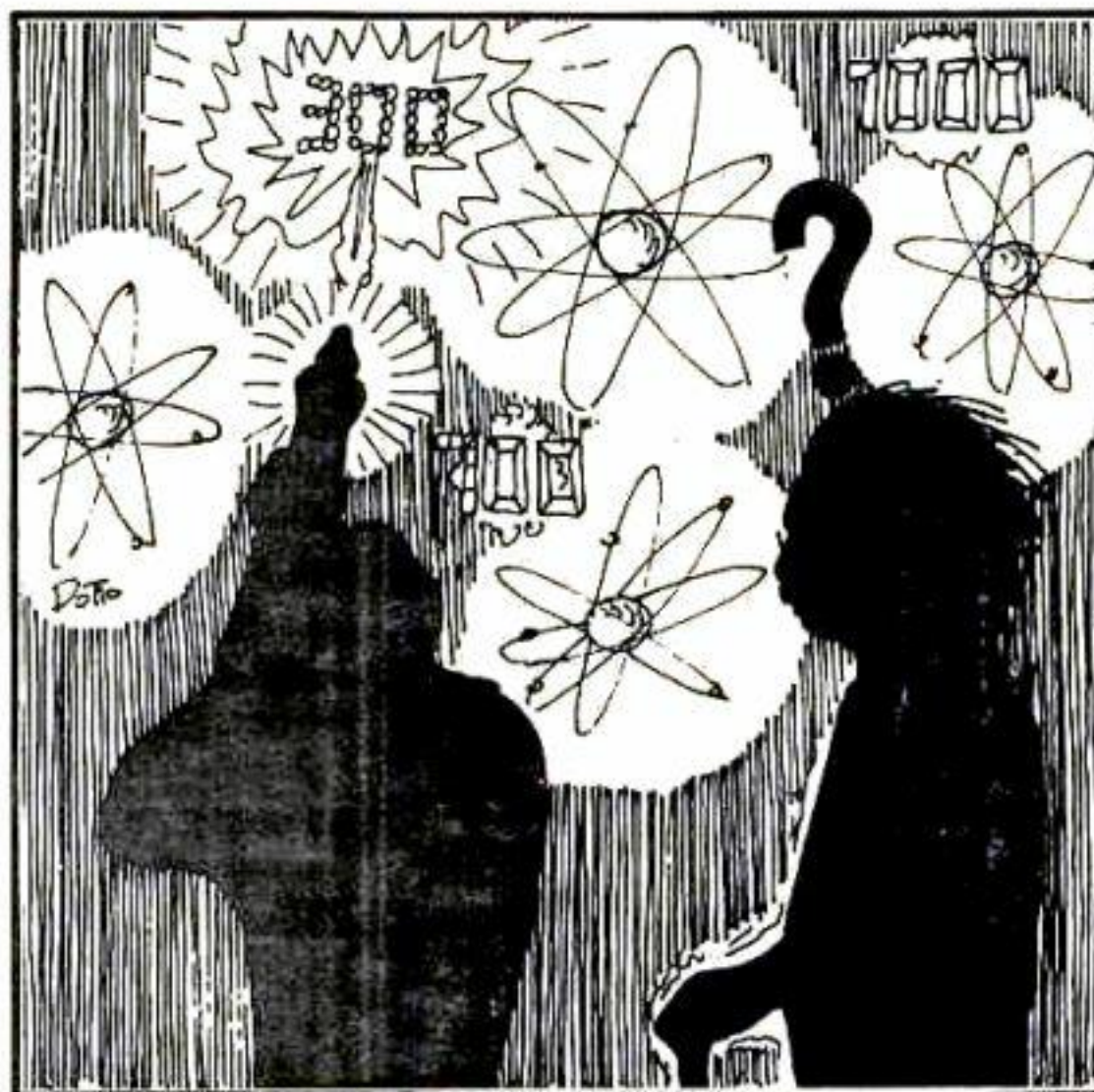
Un caso strano di compensazione solo parziale è:
X = 0.7 : IF X * 10 < > 10 * X THE PRINT X * 10 - 10 * X
in cui il THEN viene eseguito, ma il risultato è 0!

Meno pericolosi, ma comunque presenti, sono i fenomeni analoghi relativi ad interi "grandi". Ad esempio
PRINT 9 E 25
fornisce
9.00000001 E+25
mentre il ciclo:
A = 5 E 9 : FOR X = A TO A + 10 : PRINT X - A : NEXT
fornisce .. No provate a prevedervelo, poi controllate.

```
100 REM  VARIABILI IN VIRGOLA MOBI
    LE
110 CLR : A=.1
115 REM  OCCORRE CHE A SIA LA PRIM
    A VARIABILE DICHIARATA
120 B = PEEK(45)+256*PEEK(46)+2
130 REM  PEEK (X) CON X=B.....B+4
    FORNISCE V1.....V5
140 REM  CHE CORRISPONDONO ALLA VA
    RIABILE A
150 PRINT "LETTURA DIRETTA": PRINT
    "A="A
160 PRINT "VALORI V1,....V5:"
170 FOR  X=0  TO  4: PRINT PEEK(X+
    B);
180 NEXT: PRINT
190 PRINT "VALORE RICOSTRUITO=": P
    RINT "A=";
200 IF  PEEK(B)=0  THEN  PRINT 0:
    GOTO  310
210 S=1: IF  PEEK(B+1) > 127  THEN
     S=-1
220 REM  S = SEGNO
230 Z=PEEK(B+1)  OR  128:REM  RIPR
    ISTINO BIT MASCHERATO
240 Z=Z/256: FOR  X=2  TO  4: Z=Z+
    PEEK(B+X)/256↑X
250 NEXT: REM  Z= MANTISSA
260 PRINT S*Z*2↑(PEEK(B)-128)
270 PRINT "SCRITTURA BIN.NORM.": I
    F  S=-1  THEN  PRINT "-";
280 PRINT"0.";:Z=PEEK(B+1)  OR  12
    8: GOSUB  10000
290 FOR  X=B+2  TO  B+4: Z= PEEK(X
    ): GOSUB 10000: NEXT
300 PRINT "*2↑" PEEK(B)-128
310 INPUT  "VUOI CAMBIARE A (SI/NO
    )";A$
320 IF  A$="NO"  THEN  350
330 PRINT"ASSEGNA UN VALORE AD A,
    POI"
340 PRINT"ESEGUI: GOTO 150": END
350 J=0: PRINT"VUOI CAMBIARE V1...
    .V5 (S/NO)";: INPUT  A$
360 IF  A$="NO"  THEN  END
370 FOR  X=B  TO  B+4: PRINT  "V"J;
    : INPUT  Z: POKE  X,Z: J=J+1:
    NEXT: GOTO  150
380 :
1000 INPUT  Z: GOSUB  10000: END
9970 :
9980 INPUT  Z: REM  A CACCIA DI GUA
     I....
9990 :
10000 REM  TRASFORMA Z IN BINARIO AD
      OTTO BIT
10010 A$(0)= "0": A$(1)="1": A$=" "
10020 FOR  Y=1  TO  8: A$=A$(Z  AND
      1) + A$: Z=Z/2
10030 NEXT: PRINT A$;
10040 RETURN
```

Un cannone neutronico

di Antonio Visconti

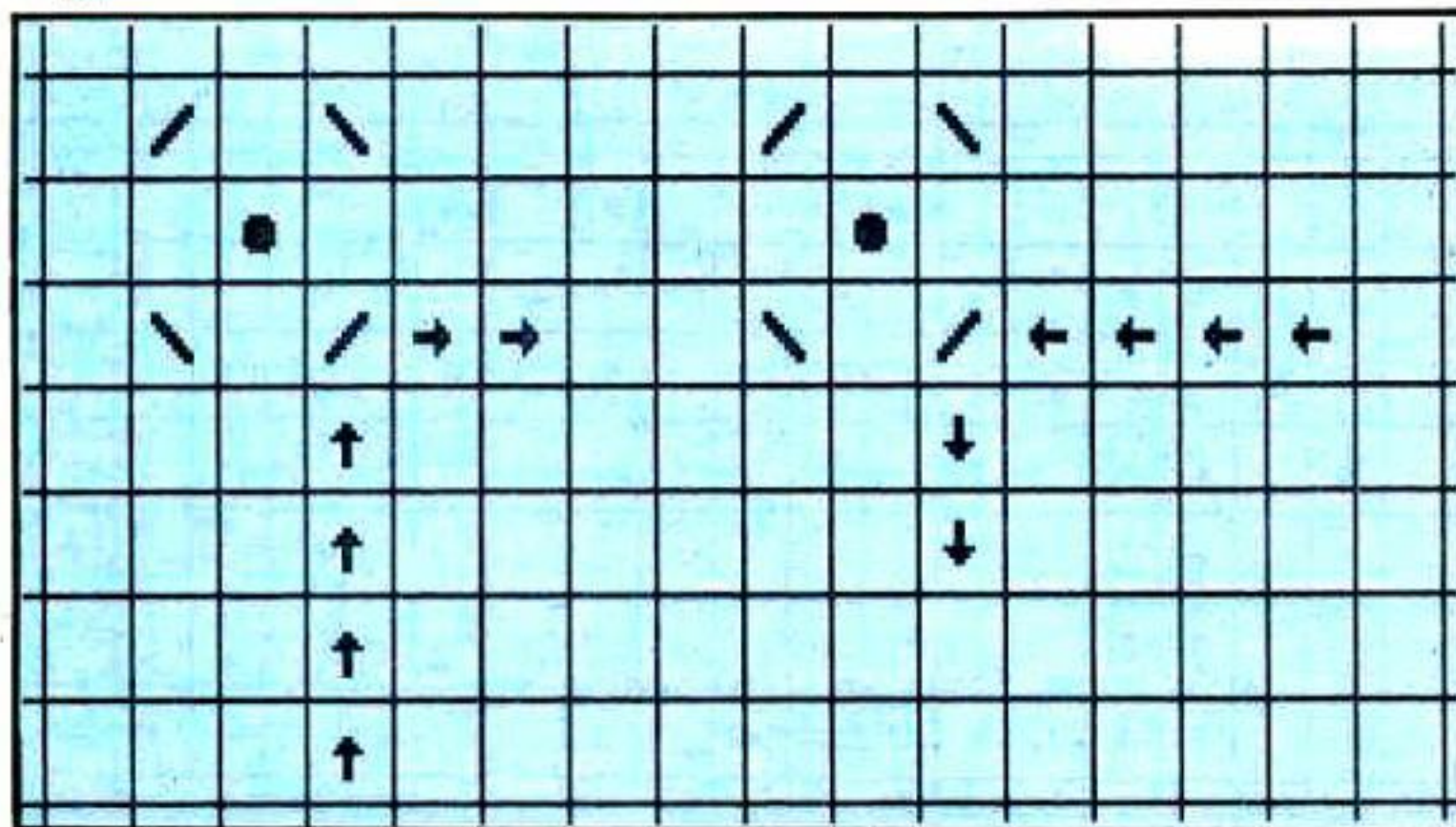
Un gioco di strategia che mette alla prova le vostre capacità di deduzione.

Nessuno ha mai osservato direttamente un elettrone, un protone, o una qualsiasi delle tante particelle atomiche scoperte negli ultimi anni. Nonostante tutto, nessuno mette in dubbio la loro esistenza: le prove sono indirette, ma comunque inequivocabili. La traccia lasciata da un elettrone sul dispositivo di rivelazione non lascia dubbi in chi è capace di interpretarla correttamente.

Il gioco

Ecco un gioco molto semplice, che vi mette nei panni di uno scienziato impegnato ad indagare sulla struttura atomica di un misterioso cristallo.

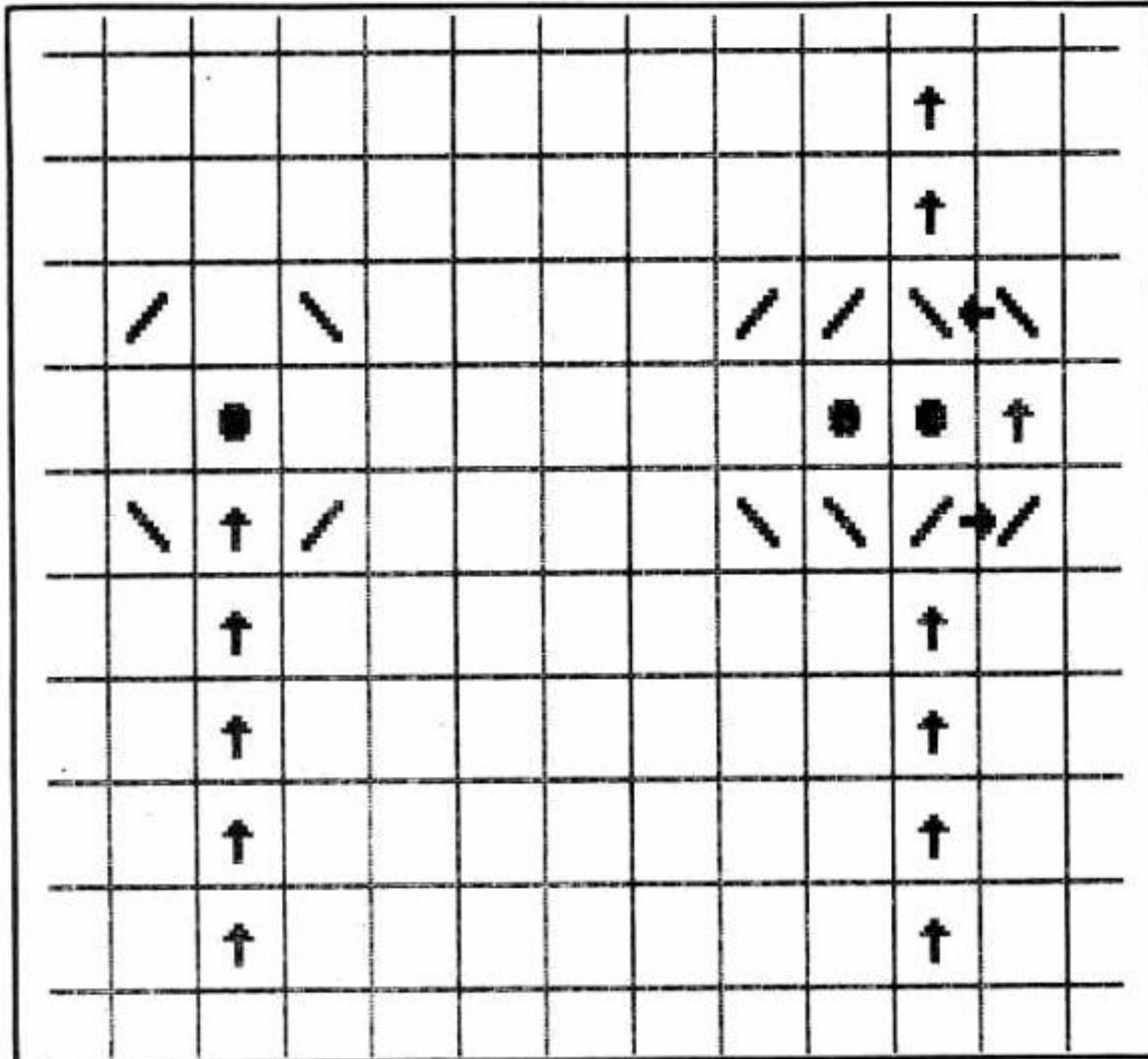
L'indagine viene fatta mediante proiettili atomici sparati violentemente all'interno del cristallo stesso. Il punto di uscita fornirà, a chi è in grado di interpretarle, utili indicazioni riguardo la posizione degli atomi che costituiscono il cristallo.

Le figure a corredo consentono di comprendere come l'effetto della collisione proiettile-atomo sia differente a seconda del punto di contatto. La figura 1, in particolare, mostra la variazione di traiettoria dovuta alla "nube elettronica". La collisione col nucleo provoca invece "l'assorbimento" (figura 2), mentre un particolare tipo di legame dà luogo ad una riflessione del proiettile (figura 3).

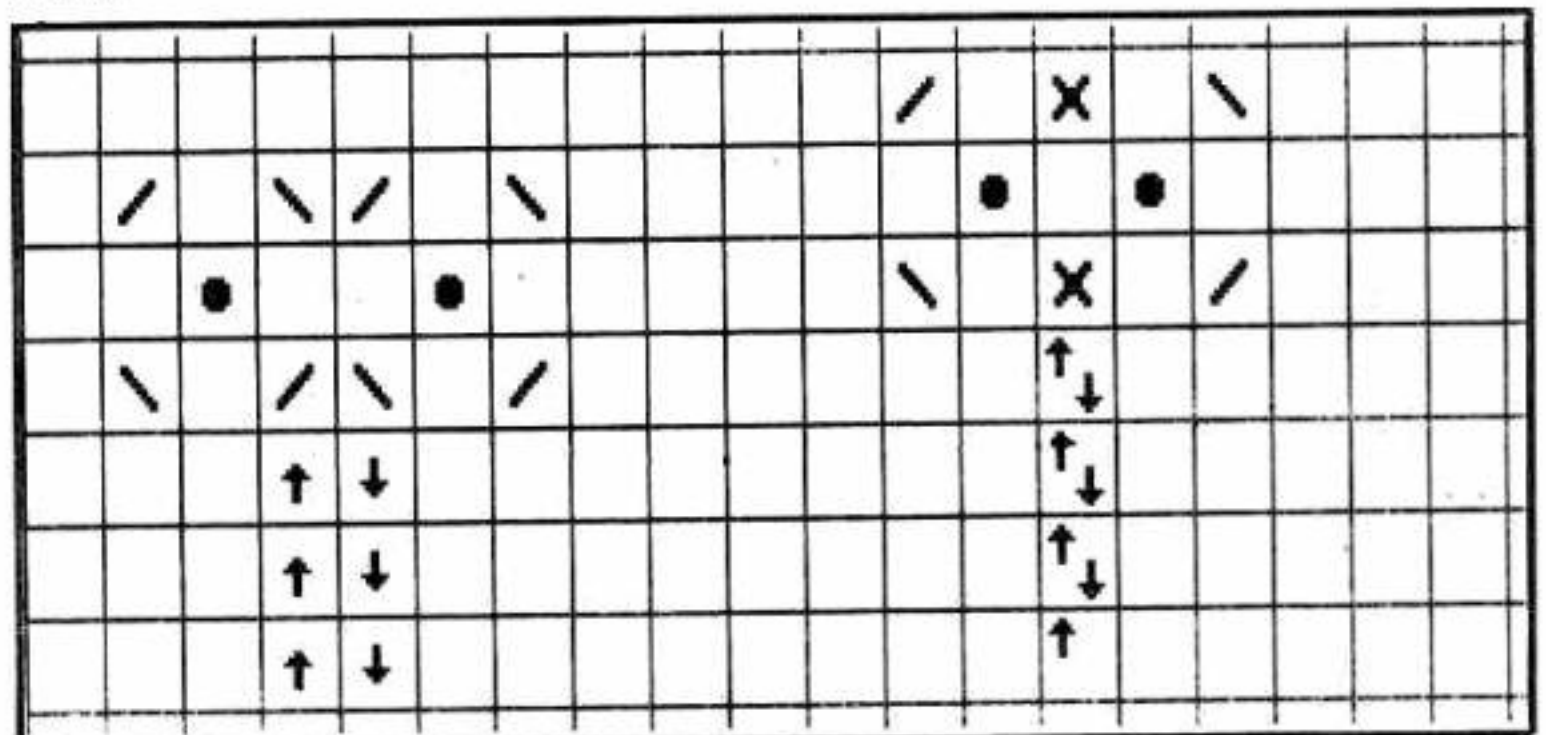
Fig. 1

Il programma

Il cristallo è rappresentato da una griglia-scacchiera di dimensioni 10 per 10; i proiettili dalle lettere dell'alfabeto. Sono previste tre opzioni, "spara", "tenta", "basta", cui si accede digitando la relativa iniziale.

Selezionando "S", un carattere blu compare sull'angolo in alto a sinistra della griglia. I tasti "<" e ">", spostano la lettera lungo la cornice e permettono di scegliere il punto di ingresso nel cristallo. Premendo la barra spaziatrice parte il colpo.

Fig. 2

Per aiutarvi nella visualizzazione, una lettera di colore bianco resterà ad indicare il punto d'ingresso, mentre un'altra di colore nero resterà ad indicare il punto di uscita. Se, lungo la cornice, un particolare carattere compare una sola volta, significherà che il proiettile è stato assorbito o riflesso a seconda che il suddetto carattere sia bianco o nero. Di notevole aiuto sono gli effetti sonori. L'urto contro un elettrone è accompagnato da un "PING", l'assorbimento da parte di un nucleo da un... "PONG". Se non viene emesso alcun suono vuol dire che il proiettile ha attraversato il cristallo, uscendo dalla parte opposta senza incontrare ostacoli.

Dopo un certo numero di colpi, se si è convinti di conoscere la posizione di un atomo, bisogna selezionare "T" (tenta) e dare le presunte coordinate. In caso di felice deduzione il punteggio si incrementa, altrimenti si ha una consistente diminuzione. Attenzione, quindi: "tentare" solo se si è sicuri. Nel caso facciate centr, i punti ottenuti sono funzione del numero di colpi sparati (più colpi = meno punti) e degli atomi già scoperti (più scoperti = più punti).

L'opzione "B" (basta) serve a chi si arrende troppo facilmente, ma è animato dalla curiosità di sapere come stanno le cose.

Il programma pubblicato è una versione unica per Commodore 64, C-16 e Plus-4.

Chi volesse modificare il commento sonoro deve alterare le apposite routine ed in particolare:

- linee 690-695: azzerano tutti i registri del SID e pongono il volume al massimo;
- linee 700-710: "PING" o "PONG" a seconda del valore di FR;
- linee 715-725: commento ad un colpo andato a vuoto;
- linee 730-740 commento ad un colpo riuscito.

La strategia

Gli atomi più superficiali sono più facili da trovare, per cui è buona norma sparare i primi colpi dagli angoli. Trovati in questo modo i primi tre o quattro, bisogna far lavorare il cervello ed avere anche un po' di... fortuna, che non guasta mai.

Non sparare da un punto da cui è uscito un proiettile perchè il percorso di andata e ritorno è univocamente determinato, per cui viene solo sprecato un proiettile. Il gioco finisce quando sono stati individuati tutti gli atomi, che sono 9, oppure finiscono i proiettili, per un totale di 26.

Fig. 3

```
100 REM COMMODORE 64
101 REM C-16 & PLUS 4
102 :
105 REM CANNONE ELETTRONICO
106 :
110 REM SIMULAZIONE SEMISERIA
115 REM DI ANTONIO VISCONTI
120 :
121 PRINTCHR$(147)"1- COMMODORE 64"
122 PRINT"2- C-16  PLUS-4"
123 GETA$:IFA$=""THEN123
124 IFA$="1"THENMS=1024:POKE53280,15:POKE53281,15
125 IFA$="1"THEN:POKE650,128:CO=64:SC=55296:GOTO130
127 MS=3072:COLOR0,2,5:COLOR4,2,5:COLOR1,2,7:SC=2048
130 FORI=1TO4:X2$=X2$+CHR$(17):NEXT:X2$=CHR$(19)+X2$
140 X3$=CHR$(19)+CHR$(17)+CHR$(17):G$=CHR$(17)
160 REM***********************************
165 DIM G(11,11),RR(9),CC(9)
170 DEF FNP(MS)=MS+40+80*C+(R+1)*2
175 PRINTCHR$(19):GOSUB500:GOSUB535
180 PRINTCHR$(19)K$CHR$(5)CHR$(18)"-RECORD"RECHR$(31)
185 GOSUB570:GOSUB690
190 PRINTX2$;K$"-SPARA":PRINT G$;K$"-TENTA":PRINTG$;K$"-BASTA"
195 PRINTCHR$(19);H$K$CHR$(145)CHR$(5)CHR$(18)CHR$(29)"PUNTI";
196 PRINT PU;CHR$(157)CHR$(146)".."CHR$(31)
200 GET A$:IF A$<>"S" AND A$<>"T" AND A$<>"B" THEN 200
205 GOSUB555
210 IF A$="T" THEN 410
215 IF A$="B" THEN PR=26
220 R=1:C=0:PR=PR+1:IF PR>26 THEN GOSUB690:FOR K=1 TO 3:GOSUB730:NEXT:GOTO620
225 POKE FNP(MS),PR:POKE FNP(SC),70
230 GET A$:F1=1
235 ON (-(A$=CHR$(44))-(A$=CHR$(46))*2-(A$=CHR$(32))*3) GOTO245,270,275
240 GOTO230
245 IF R=0 AND C<=10 THEN POKE FNP(MS),32:C=C+F1:R=-(C=11)-(C=0):GOTO265
250 IF R=11 AND C>=1 THEN POKE FNP(MS),32:C=C-F1:R=11+(C=0)+(C=11):GOTO265
255 IF C=0 AND R>=1 THEN POKE FNP(MS),32:R=R-F1:C=-(R=0)-(R=11):GOTO265
260 IF C=11 AND R<12 THEN POKE FNP(MS),32:R=R+F1:C=11+(R=11)+(R=0):GOTO265
265 POKE FNP(MS),PR:POKE FNP(SC),70:GOTO230
270 F1=-1:GOTO245
275 F2=0:POKE FNP(MS),32
280 ON F2 GOTO305,310,190
285 IF R=0 THEN POKE FNP(MS-1),PR:POKE FNP(SC-1),113:F1=1:GOTO315
290 IF R=11 THEN POKE FNP(MS+1),PR:POKE FNP(SC+1),113:F1=-1:GOTO315
295 IF C=0 THEN POKE FNP(MS-40),PR:POKE FNP(SC-40),113:F3=1:F1=1:GOTO310
300 IF C=11 THEN POKE FNP(MS+40),PR:POKE FNP(SC+40),113:F3=1:F1=-1:GOTO310
305 GOSUB320:GOTO190
310 IF F3=1 THEN F3=-1:GOSUB375:GOTO280
315 F3=1:GOSUB340:GOTO280
320 IF R<1 THEN POKE FNP(MS-1),PR:POKE FNP(SC-1),0:RETURN
325 IF R>10 THEN POKE FNP(MS+1),PR:POKE FNP(SC+1),0:RETURN
330 IF C<1 THEN POKE FNP(MS-40),PR:POKE FNP(SC-40),0:RETURN
335 POKE FNP(MS+40),PR:POKE FNP(SC+40),0:RETURN
340 R=R+F1:IF R>10 OR R<1 THEN F2=1:RETURN
345 ON G(R,C) GOTO 355,360,365,370
350 GOTO340
355 F1=-F1
360 F2=2:FR=200:GOSUB690:GOSUB700:RETURN
365 F1=-F1:FR=200:GOSUB690:GOSUB700:GOTO340
370 F2=3:FR=10:GOSUB690:GOSUB700:RETURN
```

```
375 C=C+F1:IF C>10 OR C<1 THEN F2=1:RETURN
380 ON G(R,C) GOTO 390,395,400,405
385 GOTO375
390 F1=-F1
395 F2=2:FR=200:GOSUB690:GOSUB700:RETURN
400 F1=-F1:FR=200:GOSUB690:GOSUB700:GOTO375
405 F2=3:FR=10:GOSUB690:GOSUB700:RETURN
410 PRINTX2$;K$"...DAMMI":PRINTG$;K$"     LE":PRINTG$;K$" COORDINATE"
415 FOR I=1 TO 10:POKE MS+42+2*I,I:NEXT
420 GET A$:IF A$<"A" OR A$>"J" THEN 420
425 R=ASC(A$)-64
430 FOR I=1 TO 10:POKE MS+42+2*I,32:NEXT
435 FOR I=1 TO 10:POKE MS+42+80*I,47+I:NEXT
440 GET A$:IF A$<"0" OR A$>"9" THEN 440
445 C=ASC(A$)-47
450 FOR I=1 TO 10:POKE MS+42+80*I,32:NEXT
455 GOSUB555
460 IFG(R,C)<>4THENPRINTX3$;K$"-MANCATO! ":PU=PU-950:GOSUB690:GOSUB730:GOTO190
465 F4=0:FOR I=0 TO AT:F4=F4+(C=CC(I))*(R=RR(I)):NEXT
470 IF F4<>0 THEN PRINTX3$;K$"GIA' TROVATO":GOSUB690:GOSUB730:GOTO190
475 PRINTX3$;K$;"-CENTRO!!-":PU=PU+10*(25-PR)+50*AT
480 POKE FNP(MS),81:POKE FNP(SC),70
485 PRINTCHR$(19)H$K$CHR$(145)CHR$(29)"PUNTI:"PU;CHR$(157)".."
490 GOSUB715:AT=AT+1:CC(AT)=C:RR(AT)=R:IF AT<9 THEN 190
495 NS=1:GOSUB715:GOTO620
500 C2$=CHR$(32)+CHR$(32)+CHR$(32)
501 B$=C2$+CHR$(171):FOR J=1 TO 9:B$=B$+CHR$(195)+CHR$(219):NEXTJ
505 C$=C2$+CHR$(194):FOR J=1 TO 10:C$=C$+CHR$(32)+CHR$(194):NEXTJ
510 E$=C2$+CHR$(176):FOR J=1 TO 9:E$=E$+CHR$(195)+CHR$(178):NEXTJ
515 D$=C2$+CHR$(173):FOR J=1 TO 9:D$=D$+CHR$(195)+CHR$(177):NEXTJ
520 B$=B$+CHR$(195)+CHR$(179):E$=E$+CHR$(195)+CHR$(174)
525 D$=D$+CHR$(195)+CHR$(189)
530 FOR I=1 TO 25:K$=K$+CHR$(29):NEXT:FOR I=1 TO 20:H$=H$+G$:NEXT:RETURN
535 PRINTCHR$(144)CHR$(147)CHR$(17):PRINTE$
540 FOR J=1 TO 9:PRINTC$:PRINTB$:NEXTJ
545 PRINTC$:PRINTD$
550 PRINTCHR$(31):RETURN
555 PRINTX3$;K$"             ":REM 13 SPACE
560 FOR K=1 TO 3:PRINTG$K$"------------":NEXT
565 RETURN
570 FOR I=1 TO 9
575 C%=RND(0)*8+1:R%=RND(0)*8+1
580 IF G(C%+1,R%+1)=4 THEN 575
585 G(C%+1,R%+1)=4
590 FOR X=0 TO 1:FOR Y=0 TO 1
595 A=G(C%+2*X,R%+2*Y)
600 IF A=1 OR A=2 OR A=3 THEN G(C%+2*X,R%+2*Y)=3:GOTO615
605 IF A=4 THEN 615
610 G(C%+2*X,R%+2*Y)=-(X=Y)-(X<>Y)*2
615 NEXT:NEXT:NEXT:RETURN
620 FOR C=1 TO 10:FOR R=1 TO 10
625 IF G(R,C)=1 THEN POKE FNP(MS),78:POKE FNP(SC),113:GOTO645
630 IF G(R,C)=2 THEN POKE FNP(MS),77:POKE FNP(SC),113:GOTO645
635 IF G(R,C)=3 THEN POKE FNP(MS),86:POKE FNP(SC),113:GOTO645
640 IF G(R,C)=4 THEN POKE FNP(MS),81:POKE FNP(SC),70:GOTO645
645 NEXT:NEXT
650 PRINTX2$;K$" ■■ VUOI ":PRINTK$"■ GIOCARE "
655 PRINTK$" ■ ANCORA ? "
660 GET A$:IF A$<>"S" AND A$<>"N" THEN 660
665 IF A$="N" THEN PRINT"■":END
```

```
670 IF PU>RE THEN RE=PU
675 GOSUB535:PRINTCHR$(19)K$CHR$(5)CHR$(18)".RECORD"RE;CHR$(31)
680 FOR I=0 TO 11:FOR J=0 TO 11:G(I,J)=0:NEXT:NEXT
685 FOR I=0 TO 9:CC(I)=0:RR(I)=0:NEXT:AT=0:PU=0:PR=0:NS=0:GOTO185
690 IF CO<>64 THEN VOL8:RETURN
695 FOR I=54272 TO 54295:POKE I,0:NEXT:POKE 54296,15:RETURN
700 IF CO<>64 THEN SOUND1,FR*5,10:FORI=1TO100:NEXT:RETURN
705 W=54276:POKE W+2,9:POKE W-3,FR:POKE W,23:POKE W,22
710 FOR I=0 TO 100:NEXT:RETURN
715 IF CO<>64THENFORI=1TO800STEP5:SOUND1,I,1:NEXT:RETURN
720 W=54276:POKE W+1,8:POKE W+2,251:POKE W,23:A=14:FOR I=0 TO NS:POKE54287,A
725 FOR J=1 TO 250:POKEW-3,J:A=A+.05:NEXT:NEXT:POKE W,22:RETURN
730 IF CO<>64 THEN FORI=500TO1STEP-5:SOUND1,I,1:NEXT:RETURN
735 W=54276:POKE W+1,8:POKE W+2,248:POKE W,85:I=100:J=230
740 FOR Z=1TO25:POKE W-4,I:POKE W+11,J:I=I*.98:J=J-8:NEXT:POKE W,84:RETURN
750 END
```


6

GRUPPO CONTINUITÀ

Fornito senza le 12 batterie a stilo ricaricabili. Consente il funzionamento del Vostro computer Commodore C64 o VIC 20 in assenza di corrente. Durata di funzionamento 30 minuti. Ricarica tramite alimentatore Commodore.

KIT ALLINEAMENTO TESTINA

Composto dal cacciavite, nastro di controllo e strumento di taratura con monitor audio permette il perfetto allineamento dei registratori digitali anche con nastri commerciali.

VELOCIZZATORE DI CARICAMENTO FLOPPY

Cartridge con un insieme di utility residenti su ros per velocizzare il drive nel Commodore 64.

INTERFACCIA RADIO

Indispensabile per registrare co registratore Commodore modell "C2N" i programmi speciali pe computer trasmessi dalle emitten radio.

CUFFIA PER COMMODORE C 64

Leggerissima permette l'ascolto personale del computer evitando di disturbare durante i giochi.

COPIATORE PROGRAMMI

Dispositivo hardware per effettuare copie di nastri protetti o turbo utilizzando due registratori Commodore o compatibili.

DUPLICATORE CASSETTE

Indispensabile per realizzare delle copie, con ur registratore normale, di un nastro protetto o cor caricamento turbo

Bus quadrislot	Art. CD 100	L. 55.000
Interfaccia cassette	Art. CD 101	L. 30.000
Duplicatore cassette	**Art. CD 102**	**L. 30.000**
Copiatore programmi	**Art. CD 103**	**L. 30.000**
Interfaccia radio	**Art. CD 104**	**L. 30.000**
Kit allineamento testina	**Art. CD 105**	**L. 47.000**
Alimentatore per C64 e VIC 20	Art. CD 106	L. 45.000
Gruppo continuità (fornito senza le 12 batterie a stilo ricaricabili)	**Art. CD 107**	**L. 66.000**
Pacco batterie (12 stilo 1,2 Volt ricaricabili	Art. CD 117	L. 52.000
Commutatore antenna TV/computer	Art. CD 108	L. 9.500
Tasto reset	Art. CD 109	L. 5.500
Interfaccia Centronics	Art. CD 112	L. 104.000
Espansione di memoria per C 16	Art. CD 114	L. 158.000
Velocizzatore di caricamento flop,	**Art. CD 115**	**L. 49.000**
Espansione di memoria per VIC 20 16K	Art. CD 116	L. 112.000
Modulatore Executive	Art. CD 120	L. 72.000
Penna ottica grafica	Art. CD 121	L. 45.000
Tavoletta grafica	Art. CD 130	L. 238.000
Multipresa con filtro - 2 prese	Art. CD 140	L. 41.000
Cuffia per Commodore C 64	**Art. CD 150**	**L. 19.000**
Stabilizzatore elettronico di tensione 500 W	Art. CD 160	L. 430.000
Gruppo di cintinuità 60 W	Art. CD 170	L. 400.000
Gruppo di continuità 200 V	Art. CD 180	L. 802.000
Inventer 12 Volt cc. 220 Volt ca. 100 Watt	Art. CD 190	L. 297.000
Cavo alimentazione	Art. CD 200	L. 4.600
Cavo drive o stampante Commodore	Art. CD 205	L. 8.500
Prolunga per Joystick - mt. 3	Art. CD 210	L. 25.000
Prolunga per cavo TV - mt. 3	Art. CD 215	L. 12.500
Cavo audio - mt. 6	Art. CD 220	L. 15.500
Adattatore Joistick (Atari e C64 al C 16)	Art. CD 225	L. 10.500
Adattatore registratore per C 16	Art. CD 226	L. 19.500
Mascherina antiriflesso 12"	Art. CD 300	L. 35.000
Nastro inchiostrato per Tally - mt. 80	Art. CD 610	L. 16.500
Nastro inchiostrato per Tally - mt. 180	Art. CD 611	L. 16.500
Nastro inchiostrato per Tally 1000 e Honeywell	Art. CD 612	L. 9.500
Nastro inchiostrato per Commodore MRS 801	Art. CD 614	L. 13.000
Nastro inchiostrato per Commodore MPS 802	Art. CD 616	L. 18.000
Nastro inchiostrato per Commodore MPS 803	Art. CD 618	L. 19.500
Mause per Commodore C 64	Art. CD 860	L. 240.000
Pacco carta lettura facilitata 24"×11" modulo da 500 fogli con bordi a strappo	Art. CD 630	L. 13.500
Supporto stampante porta carta in plexiglass "fume" - normale	Art. CD 660	L. 59.000
Supporto stampante porta carta in plexiglass "fume" - rinforzato	Art. CD 670	L. 80.000
Floppy disk 5" singola faccia doppia densità "ODP" - conf. 10 pezzi	Art. CD 700	L. 40.000
Floppy disk 5" singola faccia doppia densità "CBS" - conf. 10 pezzi	Art. CD 702	L. 38.000
Floppy disk 5" singola faccia doppia densità "VERBATIM" - conf. 10 pezzi	Art. CD 704	L. 42.000
Floppy disk 5" singola faccia doppia densità "DYSAN" - conf. 10 pezzi	Art. CD 706	L. 68.00(
Nastri magnetici C 10 digitali - conf. 10 pezzi	Art. CD 712	L. 20.00(
Nastri magnetici C 15 digitali	Art. CD 714	L. 21.00(
Copritastiera in plexiglass per C64 - C16 e VIC 20	Art. CD 750	L. 16.00(
Copritastiera in stoffa per C64 - C16 e VIC 20	Art. CD 760	L. 10.50(
Vaschetta portafloppy in plexiglass per 40 dischi con chiave	Art. CD 770	L. 30.00(
Vaschetta portafloppy in plexiglass per 90 dischi con chiave	Art. CD 780	L. 37.00(
Kit pulizia testine registratore	Art. CD 815	L. 13.50(
Kit pulizia disk drive	Art. CD 820	L. 26.000
Kit pulizia tastiera	Art. CD 830	L. 16.500
Foratore disk in plastica (per utilizzare la seconda faccia dei dischi)	Art. CD 840	L. 10.000
Foratore disk in metallo "tako"	Art. CD 849	L. 14.000
Joystick Spectravideo II	Art. CD 850	L. 27.000
Joystick a Microswitch	Art. CD 851	L. 52.500
Joystick senza fili con unità ricevente (funziona a batteria)	Art. CD 852	L. 98.000
Joystick per Commodore 16 (originale)	Art. CD 130	L. 29.500

TUTTI I PREZZI SONO COMPRENSIVI DI IVA
NON SI ACCETTANO ORDINI INFERIORI A L. 30.000
CONTRIBUTO FISSO SPESE DI SPEDIZIONE L. 5000

SI ACCETTANO ANCHE ORDINI TELEFONICI AI NUMERI 0522/661647-661471

BUONO DI ORDINAZIONE

NOME - COGNOME

INDIRIZZO | N.

C.A.P. | CITTÀ | PROVINCIA

VOGLIATE INVIARMI IN CONTRASSEGNO

N.	Art.	L.
N.	Art.	L.
N.	Art.	L.
SPESE SPEDIZIONE		L. 5.000
PAGHERÒ AL POSTINO		L.

COMPUTER SERVICE VIA A. MANZONI, 49 - 42017 NOVELLARA (RE) - TEL. (0522) 661647

Una stella di stampante

di **Alessandro de Simone**

Molto spesso, proprio da queste colonne, abbiamo vivamente sconsigliato i nostri lettori dall'acquistare prodotti che non fossero di marca Commodore. In molti casi, infatti, dopo alcune prove su tali apparecchi, ci rendevamo conto che la compatibilità non era reale e, nei casi più fortunati, era necessario ricorrere a programmi in linguaggio macchina che adattassero il computer alla periferica in prova.

Nel caso della stampante STAR SG-10C, invece, non si sono verificati inconveninti di sorta.

Abbiamo iniziato a provarla con piccoli programmi che verificassero la validità dei comandi speciali riportati sul libretto di istruzioni. La totale assenza di inconvenienti ci ha stimolato ad utilizzarla in unione a programmi professionali e inserendo sul retro di un Commodore 64, anche cartucce che potevano interferire sul Bus seriale di trasmissione dati. I risultati sono stati i seguenti:

> *E' possibile trovare in commercio periferiche di altre marche realmente compatibili con i computer Commodore? E, se ci sono, che vantaggio offrono rispetto alle "originali"?*

- riproduzione fedelissima, nella stampa di listati BASIC, dei caratteri speciali del colore, movimento cursore eccetera;
- funzionamento ineccepibile con i programmi di word processor Easy Script e Word Pro III, compresi i caratteri speciali di stampa in allargato e in reverse;
- funzionamento perfetto col programma Calc Result in tutte le possibili funzioni di output;

QUESTO E' UN ESEMPIO DI STAMPA NORMALE

QUESTO E' UN ESEMPIO DI NLQ
NEAR LETTER QUALITY

QUESTO E' UN ESEMPIO DI
STAMPA IN REVERSE

QUESTO E'
UN ESEMPIO
DI STAMPA
ALLARGATA

Questo e' un esempio di stampa in BUSINESS MODE

*Questo e' un esempio di stampa
con i caratteri di tipo ITALICI*

Questo e' un esempio di stampa
condensata (17 caratteri per pollice)

**Questo e' un esempio di
Stampa enfatizzata.**

• nessun problema, addirittura, col programma Print Shop che, per chi non lo sapesse, esegue perfino disegni di vario formato, tipo e dimensione in alta risoluzione sul video e in seguito li "scarica" su stampante mediante una routine di Hard Copy.

Altri programmi non sono stati provati ma è intuibile che nuove prove sarebbero risultate superflue. La presunta compatibilità, infatti, cade come una mela marcia nei casi di Hard-Copy di pagine in alta risoluzione. Anche programmi di W/P, spesso, non sono ben "compresi" da stampanti poco ortodosse.

Non è questo il caso della STAR SG-10C che, in seguito a colluttazioni con i colleghi (che la volevano a loro disposizione in Redazione), ho portato a casa mia e uso normalmente.

Avrete notato, anzi, che da un po' di tempo la qualità di stampa dei listati riportati sulle pagine di Commodore Computer Club è decisamente migliorata. Quei listati, l'avrete capito, sono stampati proprio con la STAR di cui stiamo parlando...

Le caratteristiche tecniche

La velocità (120 caratteri al secondo) è, in effetti, consistente e sottolineata dal rabbioso rumore (comunque, contenuto) della testina di scrittura. Questa, di ragguardevoli dimensioni, partecipa a dare quell'immagine di sicurezza e affidabilità che, insieme al peso della stessa stampante, consente di collocarla tra gli apparecchi di una fascia "superiore".

E' possibile utilizzare fogli singoli e pacchi di fogli perforati. La loro lunghezza (11, 12 pollici) può esser fissata da un mini deviatore. Anche l'avanzamento per rigo può esser impostato a 1/6 oppure 1/8 di pollice in modo da avere documenti più compatti.

E' ovvio che le precedenti impostazioni possono esser variate via software.

Mediante minuscoli deviatori, accessibili sul lato sinistro dell'apparecchio, è possibile, inoltre, selezionare:

• caratteri "normali" ed "italici" (cioè leggermete inclinati a destra, una specie di corsivo);

• densità normale oppure condensata (10, 17 caratteri per pollice);

• stampa normale oppure NLQ (Near Letter Quality: Qualità simile a quella di una macchina da scrivere). In questo caso la testina passa due volte sullo stesso rigo e non è più individuabile la matrice di puntini, ma ogni carattere sembra battuto da un martelletto;

• numero di periferica 4 oppure 5;

• segnalatore acustico di fine pagina e conseguente inibizione di invio caratteri. Il computer, cioè, non manda più caratteri ed attende che venga inserito un nuovo foglio.

La matrice di stampa

E' di ben 8x11 nel modo standard, 8x6 in modo grafico, 7x60 (grafica bit image) e 17x11 nei caratteri programmabili. Vi sono, come è intuibile, discendenti (anche se piccoli) con conseguente elevato grado di leggibilità da parte di coloro che sono abituati ai documenti eseguibili alla macchina da scrivere.

La sua programmabilità è piuttosto semplice e, comunque, supportata da numerosi esempi sul libretto di istruzioni che, in accordo alle cattive abitudini, è solo in inglese.

Caratteri speciali consentono la stampa in allargato (fino a quattro volte), reverse, business mode (maiuscolo minuscolo), "entrata" in modo grafico, avanzamento di rigo, nuova pagina eccetera.

Indirizzi secondari permettono la selezione del modo testo oppure grafico, la formattazione di un particolare formato di stampa, il numero di linee per pagina, i messaggi di errore eccetera.

Nessun comando, insomma, è stato trascurato non solo per rendere la stampante realmente compatibile col protocollo Commodore, ma anche per sfruttare appieno le potenzialità dell'apparecchio.

Un vantaggio non trascurabile

Se le caratteristiche tecniche della stampante STAR non vi convincono a sufficienza, basterà la seguente considerazione per far cadere gli ultimi dubbi sulla validità del prodotto: il nastro, elemento spesso trascurato dal neo-acquirente, dovrebbe, al contrario, essere uno dei primi da tenere in considerazione. Chi lavora al computer, "consuma" abitualmente grandi quantità di carta sia per stampare listati che lettere, tabelle, documenti ed altro. Le dolenti note iniziano a suonare quando giunge il momento di acquistare il nastro di ricambio.

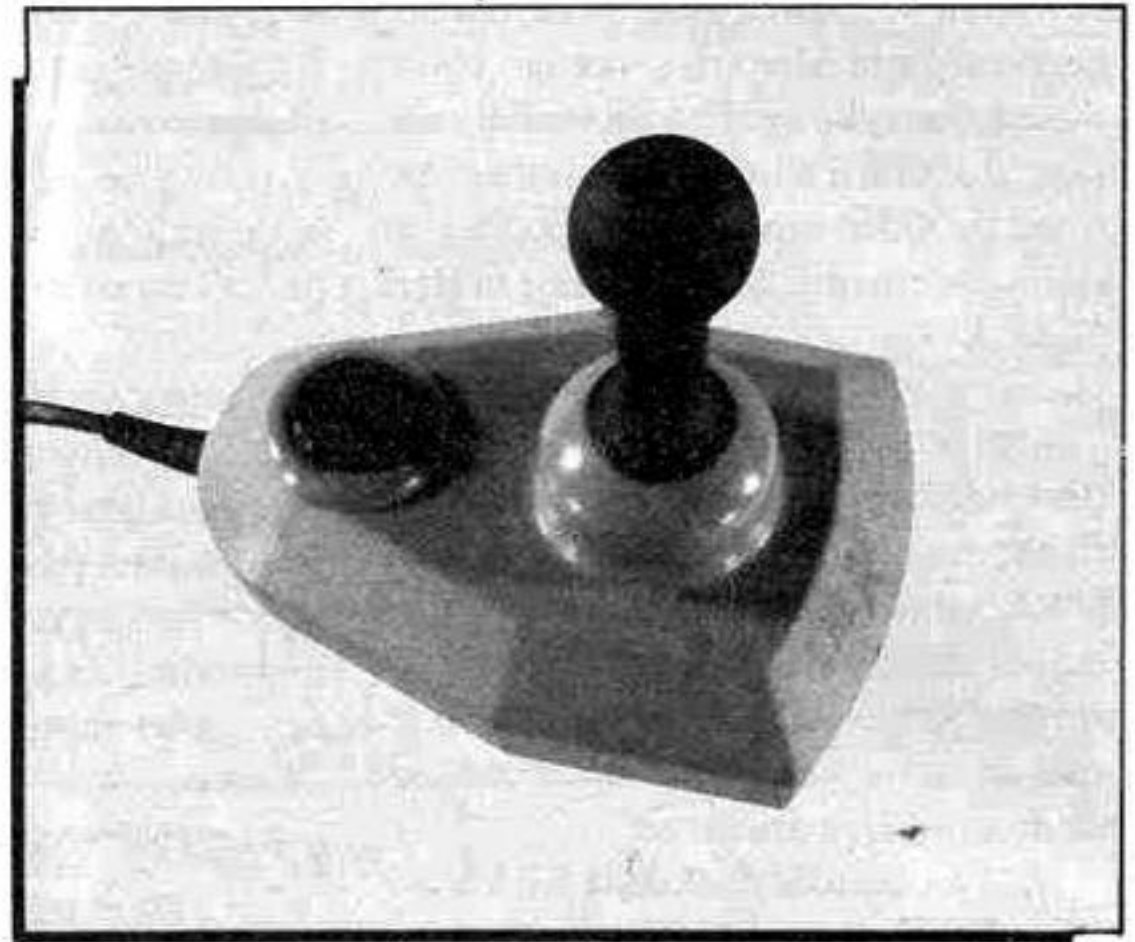
Il prezzo da pagare per l'acquisto di cartucce inchiostrate speciali, indispensabili per talune stampanti, è elevato. Se non bastasse, inoltre, troppo spesso è indispensabile acquistarle presso negozi più che specializzati, esistenti, praticamente, solo nelle grandi città.

La stampante STAR SG-10C, invece, utilizza un "normale" nastro da macchina da scrivere, rintracciabile, a prezzo contenuto, in qualsiasi cartoleria. Anche nei casi disperati in cui non si riesca a rintracciare una bobina perfettamente identica all'originale (indispensabile per il corretto inserimento nell'apposito alloggiamento), sarà pur sempre possibile estrarre il nastro da quella nuova ed utilizzare i supporti di quella vecchia.

Considerando che, per una stampante, la reperibilità di un nastro inchiostrato è vitale, il giudizio sull'apparecchio provato non può che essere positivo, anche considerando il suo prezzo di vendita che, per le prestazioni descritte, è superiore a quello di una stampante Commodore "originale".

Per ulteriori informazioni i lettori sono pregati di rivolgersi al distributore:
CLAITRON S.p.a.
Via Gallarate, 211
20151 Milano
Tel. 02/ 301.00.81

Utilizzo del joystick nel Commodore 16 e nel Commodore 64

di Diego e Federico Canetta

Il programma rappresenta una base didattica per il lettore, il quale può aggiungendovi istruzioni, può renderlo più versatile ed interessante.

Il gioco

Appena dato il RUN, sullo schermo appare il contorno entro il quale si svolge il gioco.

Appaiono poi, in posizioni casuali, alcune chiocciole, che rappresentano gli oggetti da catturare. In una zona prefissata dello schermo appare il classico "serpentello" che è possibile guidare agendo sui tasti Y, B, G, H.

Lo scopo è quello di catturare il maggior numero di chiocciole senza provocare una collisione del serpente con se stesso o con il bordo dello schermo.

Quando le chiocciole presenti sullo schermo nella prima fase del gioco sono state catturate, si passa ad uno schermo con un numero maggiore di chiocciole.

Nulla di nuovo, come potete notare, ma il breve programma sarà utilissimo per "destreggiarsi", col joystick, nei meandri del BASIC.

Il listato base

Prima di affrontare uno studio relativo all'uso del joystick, diamo insieme uno sguardo generale al programma.

Nello stendere il listato, abbiamo pensato di renderlo automaticamente adattabile ai possessori del 16 e del 64. Infatti ti, come si può vedere dalla riga 10, vi è subito un controllo del valore contenuto nella locazione 1 del calcolatore (PEEK(1)). A seconda del computer, infatti, troveremo valori diversi.

Molti lettori richiedono di illustrare come sia possibile, in un videogioco o altro programma, inserire il controllo tramite un joystick. Per risolvere il problema con il C-16 ed il C-64, ecco riproposto, un classico videogioco gestito dalla tastiera e le modifiche da apportarvi per utilizzare anche il joystick.

Se in tale locazione è contenuto il numero 200, allora il calcolatore è un 16 e si attribuisce il valore 3072 alla variabile A. Altrimenti, si passa alla riga 20 ed alla variabile A viene attribuito il valore 1024.

Questa differenziazione è resa necessaria dal fatto che i due computer, pur avendo uno schermo di dimensioni identiche, hanno le posizioni di memoria differenti. Alla variabile A viene quindi dato il valore della prima locazione dello schermo video presente in memoria, che corrisponde proprio a 3072 per il 16 e a 1024 per il 64.

Nella riga 30 viene pulito lo schermo.

Nelle righe 40 e 50 viene costruito il contorno dello schermo entro cui si svolgerà il gioco.

Nelle righe 60, 70 e 80 un ciclo FOR...NEXT si occupa di distribuire casualmente sullo schermo un certo numero di chiocciole, che corrisponderà a 5 per la prima fase del gioco (la variabile B è uguale a zero) e aumenterà di 2 unità per ogni fase successiva (vedere alla riga 110 l'istruzione B=B+2). In particolare, si noti nella riga 70 la scelta della posizione di memoria per mezzo della funzione RND(0); quindi si effettua il controllo di posizione libera in quella locazione (il valore 32 corrisponde alcarattere della barra spaziatrice). Se il controllo ha dato esito positivo, allora si passa all'istruzione successiva che stampa il carattere della chiocciola ( ), altrimenti si ritorna all'inizio della riga 70 e si ripete l'operazione.

Dopo aver dato opportuni valori alle variabili D, A, ed F (riga 90), si passa al controllo della posizione attuale del valore contenuto nella locazione di schermo davanti alla "testa" del serpente con le seguenti possibilità:

a) se in tale locazione è presente il valore 160 (riga 100) vuol dire che il serpente ha toccato se stesso od il bordo dello schermo, per cui il controllo del programma passa alla riga 180 ed il gioco finisce con l'indicazione del punteggio raggiunto;

b) se è presente il valore 0 (riga 110) allora si è in presenza di una chiocciola ed il punteggio viene aumentato (E=E+10); inoltre si incrementa la variabile F che rappresenta le chiocciole catturate, con un successivo controllo che permette di passare ad un altra fase del gioco se le chiocciole catturate sono tutte quelle presenti sullo schermo;

c) se non si è verificata nessuna delle due precedenti situazioni, allora si passa alla riga successiva.

Nella riga 120 si stampa una nuova casella colorata che rappresenta l'avanzamento del serpente; inoltre viene letto il valore della variabile A$. E' interessante notare che si è usata la funzione GET per leggere il valore di A$: in tal modo si evita l'interruzione dell'esecuzione del programma che avverrebbe utilizzando, ad esempio, l'istruzione INPUT.

La variabile A$ (variabile stringa) può corrispondere ai tasti Y, B, G oppure H, causando uno spostamento del serpente rispettivamente verso l'alto, il basso, a sinistra e a destra.

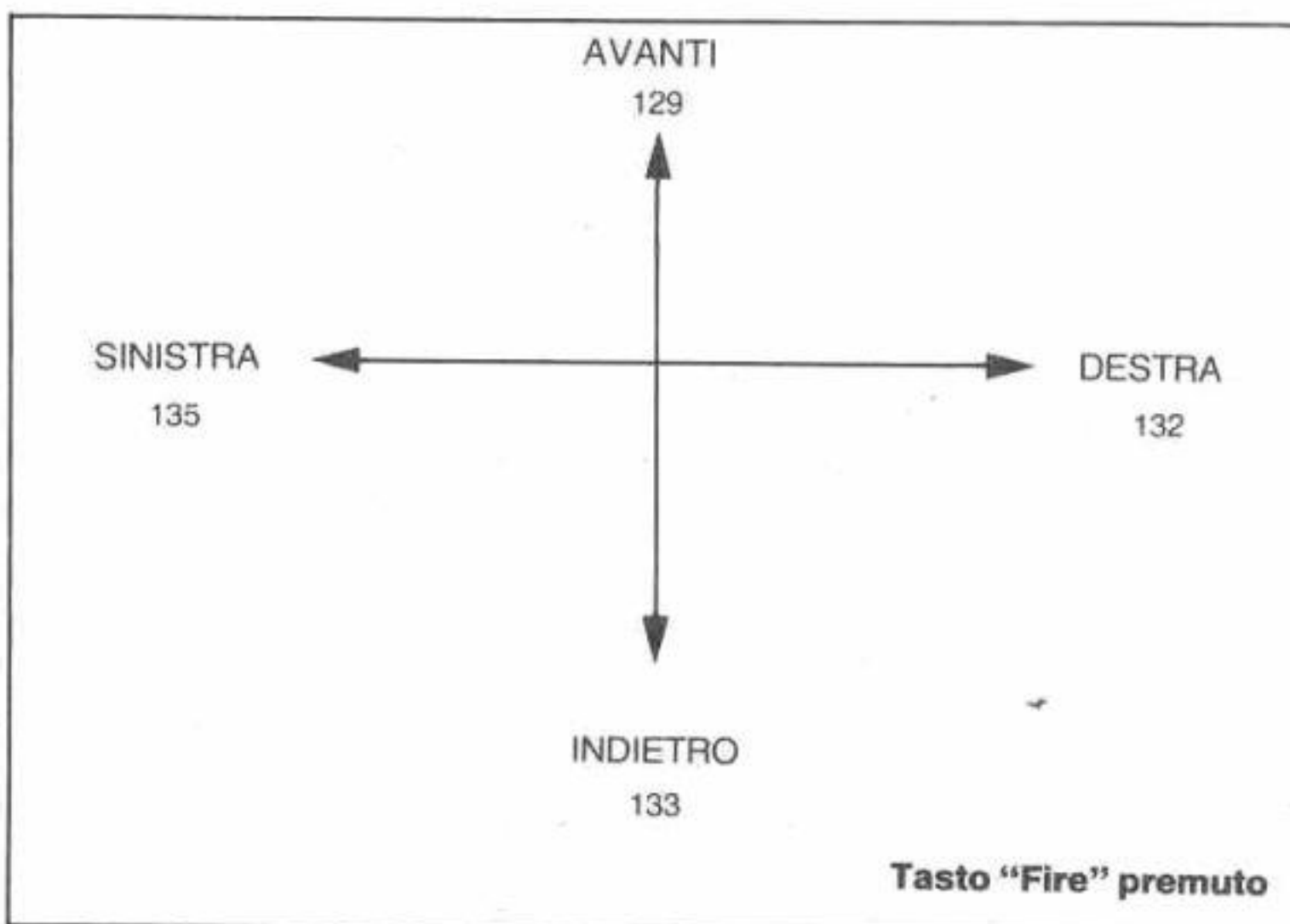



Figura 1. *Valori assunti dalle variabili JOY(1) e JOY(2) al variare della posizione della leva del joystick.*

Infine la riga 170 contiene un ciclo FOR che provoca il rallentamento del serpente, ma... attenzione: poichè è presente la variabile B, il serpente andrà più veloce in ogni fase successiva del gioco!

Controlliamo l'esecuzione del programma con il joystick

Come abbiamo detto, le righe da 130 a 160 servono a controllare il movimento del serpente per mezzo della tastiera. Se avete già provato il gioco, vi sarete però accorti che il controllo attraverso i tasti è molto meno agevole dell'uso del joystick: vedremo quindi quali sono le istruzioni che permettono di utilizzare quest'ultimo al posto dei tasti.

Nel Commodore 16 vi sono due variabili BASIC molto semplici da utilizzare, che permettono il controllo del programma tramite il joystick: JOY(1) per il joystick inserito nella porta 1 e JOY(2) per quello inserito nella porta 2.

Il valore assunto dalle variabili è dipendente dalla posizione posseduta dalla leva del joystick in quel momento: i valori corrispondenti sono riportati in figura 1.

Le istruzioni da sostituire sono talmente semplici da non richiedere commenti:

```
130 IFJOY(2) = 1THEND = -40
140 IFJOY(2) = 5THEND = 40
150 IFJOY(2) = 7THEND = -1
160 IFJOY(2) = 3THEND = 1
```

Nel Commodore 64, invece, non esistono istruzioni BASIC che permettono di controllare la posizione del joystick, ma vi sono due locazioni di memoria, una per ogni joystick, le quali contengono valori numerici differenti a seconda della posizione della leva.

Come sappiamo, ogni locazione di memoria è formata da 8 bit. Nelle locazioni 56320 e 56321 tutti gli 8 bit sono normalmente posti a 1, ma quando il

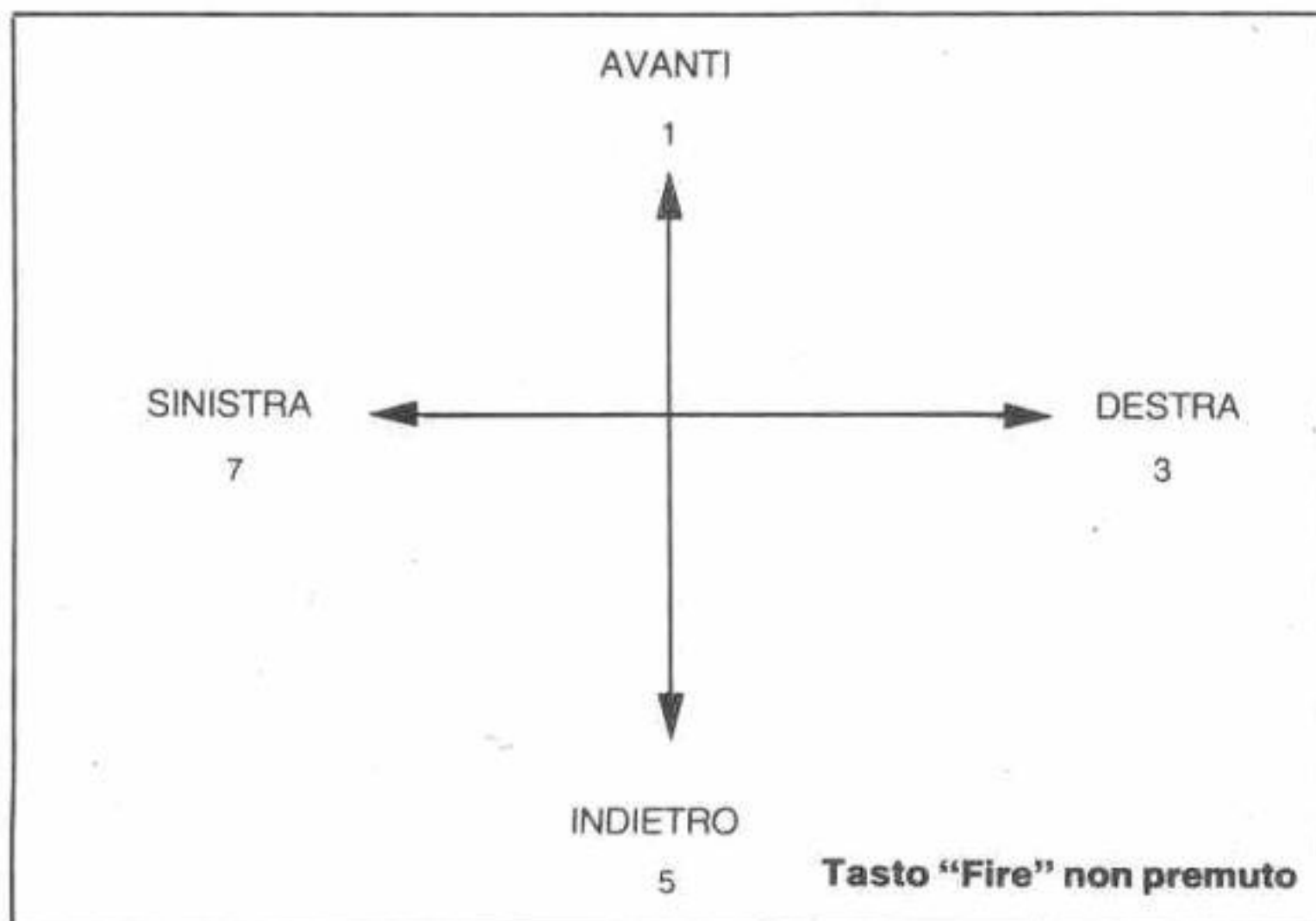
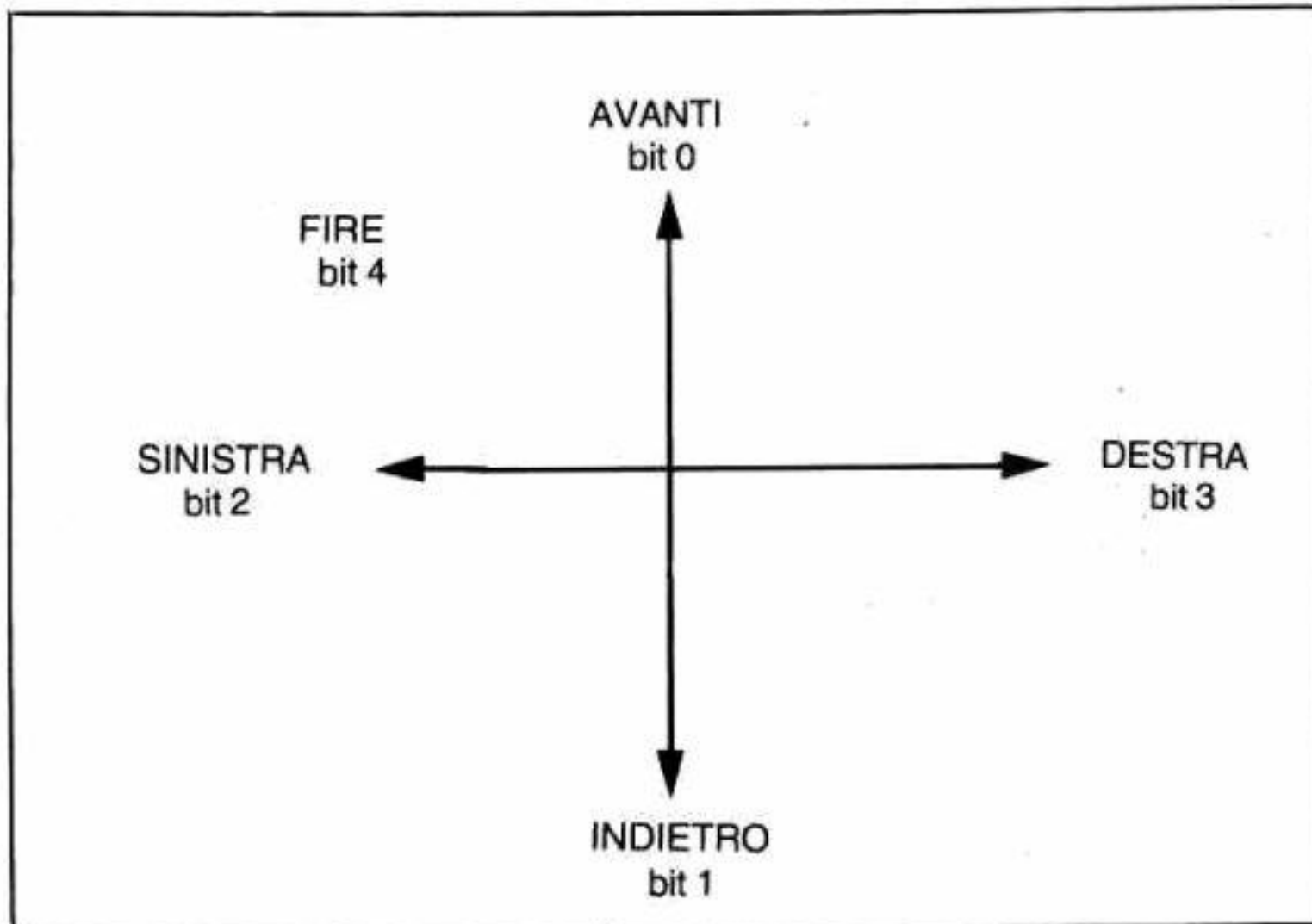



Figura 2. *Ecco quali bit delle locazioni 56320 e 56321 cambiano spostando la leva del joystick.*

joystick viene mosso, uno dei 5 bit più "bassi" assume il valore 0. In figura 2 abbiamo riportato i bit che cambiano in funzione della posizione della leva.

Per poter controllare l'esecuzione del programma tramite joystick vi sono diversi modi: uno dei più semplici è quello di ricorrere all'istruzione AND. Tale istruzione presenta qualche difficoltà di carattere concettuale, per cui ai lettori alle prime armi consigliamo di evitare la spiegazione data di seguito ed utilizzare i dati contenuti in tabella 1 oltre agli esempi che seguono per utilizzare il joystick.

Ci limiteremo a dire che l'istruzione AND permette di confrontare tra loro due numeri nelle rappresentazione binaria. Possono verificarsi i seguenti casi:

0 AND 0 = 0
0 AND 1 = 0
1 AND 0 = 0
1 AND 1 = 1

Come si vede dagli esempi, l'istruzione AND fornisce come risultato il valore 1 solo se si confrontano due valori uguali ad 1. Vediamo ora un esempio su 8 bit:

0000111 (= 7 in decimale)
0001010 (= 10 in decimale)

0010001 (= 17 in decimale)

La funzione AND può ovviamente essere utilizzata anche per confrontare direttamente due numeri memorizzati durante un'elaborazione.

Riferendosi alla tabella 1, le istruzioni da sostituire nel listato pubblicato, valide per il COMMODORE 64, sono le seguenti: (joystick in porta 2)

```
130IF(PEEK(56320)AND1)
=0THEND=-40
140 IF(PEEK(56320)AND2)
=0THEND=40
150 IF(PEEK(56320)AND4)
=0THEND=-1
160 IF(PEEK(56320)AND8)
=0THEND=1
```

Attenti a digitare bene le parentesi!

Nel caso in cui la leva del joystick occupi la posizione centrale, tutti gli IF non sono verificati, poichè i bit della locazione 56320 sono "normalmente" posti a 1. Se la leva è spinta in avanti, il primo bit viene posto a 0 e l'istruzione IF alla riga 130 viene verificata, per cui si ha D=-40.

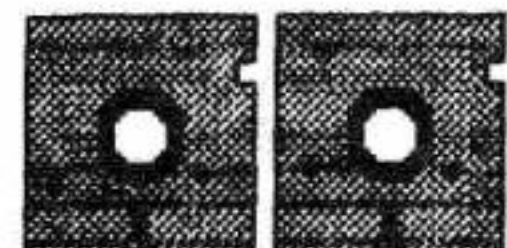
Miglioramento del gioco

Il listato che abbiamo presentato è molto semplice e, come abbiamo detto all'inizio, rappresenta un punto di partenza per elaborazioni più o meno complesse. Le modifiche che possono essere effettuate sono sia di carattere estetico, preparando ad esempio una buona presentazione iniziale con istruzioni del tipo PRINT, sia di carattere funzionale che rendano il gioco più elaborato. Ad esempio:

- memorizzate il punteggio finale del gioco che rappresenta il "record" raggiunto. Il record va aggiornato ad ogni fine partita.
- oltre alle chiocciole, inserite altre figure che diano punteggi diversi od ostacoli che vadano accuratamente evitati.
- per i più esperti, inserite una subroutine sonora che produca un suono ad ogni collisione con una chiocciola.

Con il programma proposto speriamo di essere venuti incontro a quei lettori a cui non basta giocare, ma pretendono un uso più creativo del computer.

VALORE	POSIZIONE LEVA
1	AVANTI
2	INDIETRO
4	SINISTRA
5	AVANTI-SINISTRA
6	INDIETRO-SINISTRA
8	DESTRA
9	AVANTI-DESTRA
10	INDIETRO-DESTRA
16	FIRE
17	AVANTI-FIRE
18	INDIETRO-FIRE
20	SINISTRA-FIRE
21	AVANTI-SIN.-FIRE
22	IND.-SIN.-FIRE
24	DESTRA-FIRE
25	AVANTI-DESTRA-FIRE
26	IND.-DESTRA-FIRE

Tabella 1. *Valori calcolati con l'istruzione AND per il Commodore 64.*

```
  1 REM   IMPARIAMO AD
  2 REM   INSERIRE I COMANDI
  3 REM   RELATIVI AL JOYSTICK
  4 REM   C-16 E COMMODORE 64
  5 :
  6 REM   MINI-GIOCO DI FEDERICO CA
     NETTA
  7 :
 10 IF PEEK(1)=200 THEN A=3072:GOT
     O 30:REM  PER COMMODORE 16
 20 A=1024:REM  PER COMMODORE 64
 30 PRINTCHR$(147)"TASTI:Y,G,B,H":
     FOR T=1 TO 1000:NEXT:PRINTCHR$
     (147)
 40 FOR T=0 TO 39:POKE A+T,160:POK
     E A+960+T,160:NEXT
 50 FOR T=0 TO 1000 STEP 40:POKE A
     +T,160:POKE A+T-1,160:NEXT:POK
     E A+897,160
 60 FOR T=0 TO 5+B
 70 C=A+RND(0)*900:IF PEEK(C)<>32
     THEN 70
 80 POKE C,0:NEXT
 90 D=1:A=A+897:F=0
100 IF PEEK(A+D)=160 THEN 180
110 IF PEEK(A+D)=0 THEN E=E+10:F=F
     +1:IF F=6+B THEN FOR T=0 TO 50
     0:NEXT:B=B+2:GOTO 10
120 A=A+D:POKE A,160:GET A$
130 IF A$="Y" THEN D=-40
140 IF A$="B" THEN D=40
150 IF A$="G" THEN D=-1
160 IF A$="H" THEN D=1
170 FOR T=0 TO 50-B:NEXT:GOTO 100
180 PRINTCHR$(147)"FINE DEL GIOCO"
     :PRINT:PRINT"HAI FATTO"E"PUNTI
     "
```

VINCE CHI LEGGE.

Parte l'Operazione Fedeltà Systems. I premi?
Tanti: una moto Cagiva Electra 125, printer-plotter e stampanti a margherita Commodore, abbonamenti alla tua rivista Systems preferita e libri della biblioteca informatica Systems. Partecipare è semplicissimo.

Raccogli i bollini dell'Operazione Fedeltà Systems che troverai

OPERAZIONE

da oggi fino al 31 marzo 1986 - su Commodore Computer Club, Computer, Commodore, Commodore Club, MSX, Personal Computer, Sinclair Computer, 16/48, Special Systems, VR Videoregistrare e sui libri Systems.

incollali sulla cartolina inserita in ogni rivista e spediscila. Ogni mese parteciperanno all'estrazione tutte le cartoline con almeno 5 punti. E ricor-

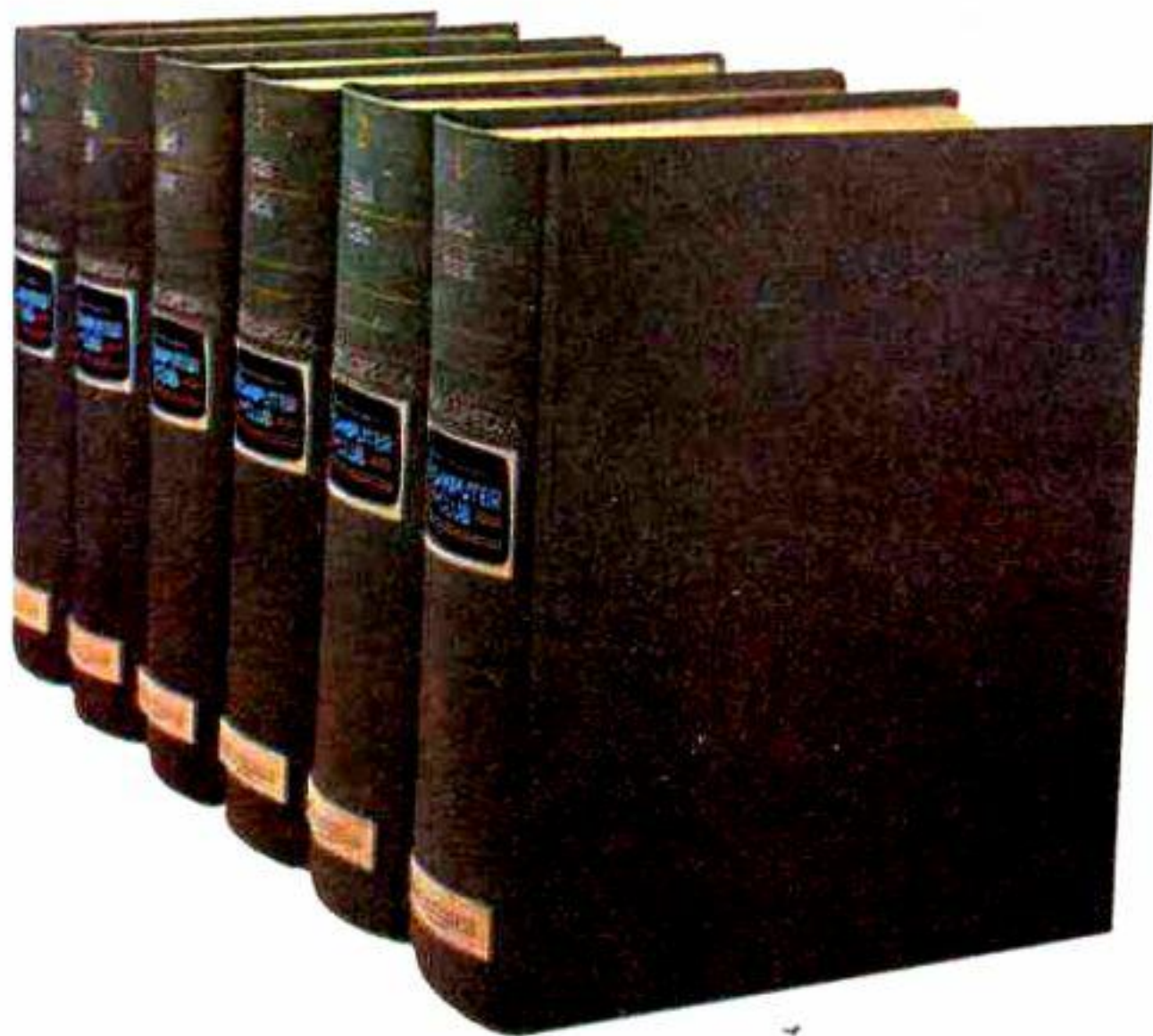
da: l'abbonamento a qualsiasi rivista Systems vale subito 5 punti. Inoltre tutte le cartoline parteciperanno al grande concorso finale: in palio una potente moto Cagiva Electra 125.

REGOLAMENTO DEL CONCORSO

Partecipano all'estrazioni mensili dell'OPERAZIONE FEDELTA' SYSTEMS tutte le cartoline pervenute con almeno 6 punti.
L'abbonamento a una qualsiasi rivista vale 5 punti subito.
I bollini che compaiono su Computer, Commodore Club, 16/48, VR Videoregistrare e i libri Systems valgono 2 punti; quelli di Commodore Computer Club, Commodore e Sinclair Computer 1 punto; quelli delle cassette Special Systems valgono 3 punti.
Il monte premi delle estrazioni mensili è così ripartito:
- dal 1° al 5° estratto: 1 stampante Commodore.
- dal 6° al 10° estratto: 1 printer plotter Commodore.
- dal 11° al 40° estratto: un libro a scelta della biblioteca informatica Systems oppure a scelta un abbonamento ad una rivista Systems.
Tutte le cartoline inviate partecipano all'estrazione finale di una moto Cagiva Electra 125 cc.

EDITORIALE SYSTEMS
"Operazione Fedeltà"
V.le Famagosta, 75
20142 Milano

Enciclopedia di routine

a cura di **Alessandro de Simone**

Come realizzare l'enciclopedia ed utilizzarla nei propri listati

Ai lettori che hanno acquistato per la prima volta questo numero di Commodore Computer Club, illustriamo qui di seguito, i vantaggi che derivano dalla raccolta.

Questa, a pensarci bene, è la versione "superiore" della rubrica "1 riga" e potrebbe anche denominarsi... "una schermata"!

Oltre che utili per costituire un'enciclopedia, i brevissimi sottoprogrammi pubblicati su ogni numero, sono anche validissimi strumenti di studio per coloro che desiderano approfondire le proprie conoscenze del basic, esaminando, senza fatica, particolari routine o insolite tecniche di programmazione.

- Dato che può esser "chiamata" più di una volta nel corso di un programma, nessuna routine contiene istruzioni del tipo DATA oppure DIM, allo scopo di non creare confusione col listato principale.
- Nessuna routine può far riferimento ad altre routine dell'enciclopedia stessa.
- Nessuna routine può contenere variabili banali (A, A$, eccetera), ma solo variabili poco usate (X1$, X8, Y0%, eccetera).
- Ogni routine deve apparire, *per intero*, sullo schermo del computer e consentire, proprio per questo motivo, di essere esaminata comodamente.
- Ogni routine deve esser numerata secondo uno standard che ha la particolarità di esser ricordato facilmente:

Righe	*Contenuto*
XXY00	Prima riga del sottoprogramma
XXY89	Ultima riga utile del sottoprogramma
XXY90	REM Prima riga di spiegazioni
XXY99	REM Nome della subroutine

in cui XX sono due valori variabili da 10 a 63; Y è un carattere numerico compreso tra 0 e 9.

Qualsiasi subroutine, in altre parole, inizia con un numero, di cinque caratteri, che termina *sempre* con "00". La stessa subroutine, d'altra parte, ha l'ultima riga numerata con "99".

Digitando, ad esempio:

LIST 10800-10899

si avrà la certezza di veder apparire sullo schermo, *per intero*, la routine il cui nome si trova nella riga 10899.

Prima di accedere alla routine è necessario assegnare, alle variabili indicate con REM da riga XXY89 a XXY98, particolari valori per il suo corretto funzionamento. Al "ritorno" una o più variabili conterranno il risultato dell'elaborazione.

In questo modo, per esser più chiari, è possibile simulare alcuni comandi di versioni basic avanzate oppure, addirittura, creare nuove e inedite istruzioni.

Ad esempio, il comando:

SOUND 1,800,500

che, nel C-16, riproduce un suono di tonalità 800 tramite la voce 1 per la durata 500, potrebbe venir riprodotta, in un'ipotetica subroutine per il Commodore 64, con:

X1=1:X2=800:X3=500:GOSUB12400

nell'ipotesi, ovviamente, che la routine in oggetto sia allocata da riga 12400 a 12499.

I listati pubblicati "girano" su ogni computer salvo dove indicato diversamente.

E' ovvio che nel caso del VIC-20, (che, come è noto, ha uno schermo di soli 506 caratteri), le subroutine "universali" funzionano correttamente, ma non possono apparire per intero in una sola schermata.

Per quanto riguarda la digitazione, si tenga presente che sulla rivista, per motivi di chiarezza, i comandi e le istruzioni basic sono separati tra loro da spazi bianchi. Nel digitare le linee di programma, pertanto, è opportuno ignorarli, per non si rischia di non restare in una sola schermata. Se, per esempio, leggete:

12100 X1=34: X2 = SQR(X3) + LOG(X1)

digitate nel modo seguente:

12100 X1=34:X2=SQR(X3)+LOG(X1)

senza, cioè, alcuna separazione tra comandi ed istruzioni.

11400 Istogrammi

(Per qualsiasi Commodore)

Una routine decisamente "povera" che consente, tuttavia, di visualizzare con la massima semplicità, sullo schermo, otto dati sotto forma di istogrammi orizzontali.

Si è fatto ricorso ai caratteri semigrafici Commodore e, in particolare, agli otto caratteri che consentono di stampare, in una singola cella video, da una a otto barrette verticali contigue. In questo modo, infatti, è possibile visualizzare anche porzioni di cursore con l'accuratezza di un ottavo.

Provate a far girare il semplice *demo* allegato. Alle domande sui valori da introdurre rispondete, ad esempio, con 10, 10.5, 11, 11.5, 12, 12.5 e zero (0) alle rimanenti. Alle domande relative ai nomi rispondete PRIMO, SECONDO, TERZO... oppure come più vi aggrada.

Vedrete apparire, dopo la digitazione dell'ultimo dato, sei istogrammi orizzontali sormontati dal nome e dal valore digitato corrispondente. All'estremità destra di ciascuna barra compare un numero, compreso tra 1 e 8, che ha il compito di evidenziare il carattere semigrafico presente alla stessa estremità.

La ricerca del valore massimo è automatica per cui, in ogni schermata, vi sarà sempre una barra che occuperà l'intera larghezza di schermo. Questo accorgimento è stato adottato per sfruttare al massimo le potenzialità di visualizzazione del video.

Non è possibile digitare valori negativi (vengono subito trasformati in positivi) come pure almeno uno dei dati deve essere non nullo. In caso contrario, come di consueto, la routine non viene eseguita e la variabile X0$ conterrà il messaggio "ERR".

A. de Simone

```
   90 REM  ESEMPIO D'USO
   92 REM  ISTOGRAMMI
   94 REM  QUALSIASI COMPUTER
   96 REM  (PER VIC-20 CAMBIARE 39
      IN 21)
   97 :
  100 FOR I=1 TO 8:INPUT "VALORE";
      X1(I)
  110 INPUT "NOME";X1$(I):NEXT:GOS
      UB 11400:PRINTCHR$(147)
  112 IF X0$="ERR" THEN PRINT"ERRO
      RE NEI DATI IMMESSI"
  115 GOTO 100
  120 :
11400 X0$="":X1(0)=0:X1(10)=0:FOR
      X1=1 TO 8:X1(I)=ABS(X1(X1)):
      NEXT:FOR X1=1 TO 9
11405 FOR X1=1 TO 8:X1$(X1)=LEFT$(
      X1$(X1)+"............",12):N
      EXT:X1$="1234567890"
11410 FOR X1=1 TO 8:IF X1(X1)<X1(0
      ) THEN X1(0)=X1(X1)
11411 NEXT:FOR X1=1 TO 8:IF X1(X1)
      >X1(10) THEN X1(10)=X1(X1)
11415 IF X1(10)=0 THEN X0$="ERR":R
      ETURN
11420 NEXT:PRINTCHR$(147);:X1=39/X
      1(10):FOR X2=1 TO 8:PRINTX2;
      X1$(X2)X1(X2)
11430 FOR X3=1 TO X1(X2)*X1:PRINTC
      HR$(18)CHR$(32);:NEXTX3:X4=X
      1(X2)*X1
11440 IF (X4-INT(X4))*10>=1 THEN P
      RINTCHR$(146)MID$(X1$,(X4-IN
      T(X4))*10,1);
11445 IF X1(X2)=0 THEN PRINTCHR$(1
      57)CHR$(146);
11450 PRINT:PRINT:NEXTX2
11460 GET X2$:IF X2$="" THEN 11460
11470 RETURN
11492 REM  X1(1)....X1(9) DATI
11493 REM  X1$(1)....X1$(9) NOMI
11499 REM  ISTOGRAMMI
```

Collaborazione dei lettori

La collaborazione dei lettori è gradita purchè si provveda ad inviare *almeno* tre sottoprogrammi per volta, su nastro, disco oppure output di stampante. I listati di routine che non rispettano lo standard adottato rischiano di non esser presi in considerazione.

Tutti i lavori pubblicati verranno compensati con prodotti della Systems Editoriale (cassette di programmi, libri, abbonamenti, copie arretrate, eccetera).

11500 Caratteri in alta risoluzione

(Solo per C-64)

Questa routine, utilizzata con le Routine Grafiche pubblicate sul N.14 di C.C.C, permette di visualizzare sulla pagina in alta risoluzione qualsiasi frase o messaggio.

Prima di eseguire il GOSUB 11500, è sufficiente assegnare la frase alla variabile stringa X1$, indicare la riga (X6) e la colonna (X5) in cui si desidera posizionare la scritta.

La variabile - errore X0$ sarà nulla se tutto è in ordine, altrimenti conterrà "ERR". Questo caso si verifica quando la variabile X1$ è nulla e quando il messaggio da visualizzare non può esser contenuto nei limiti dello schermo.

G. Castagna

```
100 REM ESEMPIO D'USO
110 REM CARATTERI IN H.RES.
120 REM SOLO PER C 64
130 REM INSERIRE LE ROUTINE
135 REM GRAFICHE DI C.C.C. N.14
140 :
150 ←GRAF12,0:←COLOR1:←CLEAR
160 ←CIRCLE0,0,0,100,100
170 X5=9:X6=20:X1$="COMMODORE COMPUTER CLUB"
180 GOSUB11500:IFX0$="ERR"THENPRINTX0$:END
190 STOP
200 :
11500 X0$="":IFX1$=""ORX6<0ORX6>24ORX5<0ORX5>39THENX0$="ERR":RETURN
11503 IFX6*40+X5+LEN(X1$)>1000THENX0$="ERR":RETURN
11505 POKE56334,PEEK(56334)AND254:POKE1,PEEK(1)AND251
11510 X0=57344+X5*8+X6*320:X1=LEN(X1$)
11520 FORX2=1TOX1:X2$=MID$(X1$,X2,1)
11525 X3=ASC(X2$):IFX3>63THENX3=X3-64
11530 FORX4=0TO7:POKE(X4+X0),PEEK(53248+X4+X3*8):NEXT:X0=X0+8:NEXT
11550 POKE1,PEEK(1)OR4:POKE56334,PEEK(56334)OR1:RETURN
11590 REM X1$= FRASE DA VISUALIZZARE IN HI-RES
11591 REM X6=NUMERO RIGA. X5=NUMERO COLONNA
11599 REM CARATTERI IN HI-RES
```

11600 Scomposizione in sillabe

(Per qualsiasi Commodore)

Un listato già proposto su pubblicazioni SYSTEMS. Modificato e compattato diventa un'utilissima routine. "Scomposizione in sillabe" rappresenta, anche da solo, un miniprogramma utilizzabile a fini didattici.

Alla domanda:

VOCABOLO?

potete rispondere sia con una singola parola che con un'intera frase (Y$), lasciando spazi vuoti tra un vocabolo e l'altro. Le sillabe (non più di 11) vengono individuate e assegnate al vettore Y$.

Variabili utilizzate: X1, Y$, X0, X0(), Y$(), X$

Nel caso vi siano più di undici sillabe, caso rarissimo, sono ignorate quelle in più.

(F. Molinari)

```
100 REM  ESEMPIO D'USO
110 REM  SCOMPOSIZIONE IN SILLAB
    E
115 REM  PER QUALSIASI COMPUTER
120 :
130 INPUT "VOCABOLO";Y$
140 GOSUB 11600:FOR I=0 TO 10:IF
     Y$(I)="" THEN I=10

145 PRINT"SILLABA"I;CHR$(18)Y$(I
    ):NEXT:GOTO 130
150 :
9999 END
11600 X1=0:FOR X9=0 TO 10:Y$(X9)="
      ":NEXT:X9=0
11605 X1=X1+1:IF X1>LEN(Y$) OR X9>
      10 THEN RETURN
```

```
11610 FOR X0=0 TO 2:X$(X0)=MID$(Y$
      ,X1+X0,1):X$=X$(X0)
11615 X(X0)=X$="I" OR X$="U" OR X$
      ="O" OR X$="A" OR X$="E":NEX
      T:IF X(0)=0 THEN 11675
11625 IF X(1)=0 THEN 11655
11635 IF X$(0)="I" OR X$(0)="U" TH
      EN 11675
11640 GOTO 11680
11650 X3=(X(2)=0)+2:ON X3GOTO 1007
      5,11680
11655 IF X(2)=-1 THEN 11680
11660 IF X$(1)=X$(2) THEN 11670
11665 IF X$(1)="S" OR X$(1)="G" OR
       X$(2)="R" OR X$(2)="L" OR X
      $(2)="H" THEN 11680
11670 Y$(X9)=Y$(X9)+X$(0)+X$(1):X1
      =X1+1:X9=X9+1:GOTO 11605
11675 Y$(X9)=Y$(X9)+X$(0):GOTO 116
      05
11680 Y$(X9)=Y$(X9)+X$(0):X9=X9+1:
      GOTO 11605
11690 REM  X$(), Y$() X$,X0,X1,X3,
      X9
11699 REM  SCOMPOSIZIONE IN SILLAB
      E
```

11700 Microcalcolatrice

(Per qualsiasi Commodore)

Consente, interrompendo momentaneamente l'esecuzione del programma principale, di effettuare semplici calcoli matematici: somme, differenze, motiplicazioni, divisioni, quadrati, radici.

E' talmente semplice che non c'è bisogno di commenti. Quando compare il menu, scegliete l'operazione che volete effettuare premendo il tasto corrispondente e Return. Battendo in INPUT i numeri (oppure il numero, se trattasi di quadrato o estrazione di radice), avrete il risultato preceduto dal simbolo corrispondente all'operazione effettuata. Nel caso venga richiesta un'operazione impossibile (divisione per zero, radice di un numero negativo) compare un opportuno messaggio.
Banale o utilissima? Fate voi.

Se proprio avete voglia di divertirvi, ricavatene un piccolo programma e proseguite nella lista delle operazioni: radici cubiche, logaritmi, seni, tangenti, fattoriali... e chi più ne ha più ne metta!

F. Molinari

```
  100 REM  ESEMPIO D'USO
  110 REM  MICROCALCOLATRICE
  120 REM  QUALSIASI COMPUTER
  130 :
  140 GOSUB 11700
  150 :
 9999 END

11700 PRINTCHR$(147):PRINT"1 ADDIZ
      IONE":PRINT"2 SOTTRAZIONE":P
      RINT"3 DIVISIONE"
11705 PRINT"4 MOLTIPLICAZIONE":PRI
      NT"5 RADICE":PRINT"6 QUADRAT
      O":PRINT"7 FINE"
11710 PRINT:PRINTX9"RISULTATO: "X3:
      PRINT:INPUT "QUALE";X9:IF X9
      <1 OR X9>7 THEN RETURN
11712 IF X9=7 THEN RETURN
11713 IF X9>4 THEN INPUT "VAL";X1:
      GOTO 11755
11715 INPUT "PRIMO VALORE";X1
11720 INPUT "SECON.VALORE";X2
11730 IF X9=1 THEN X3=X1+X2:GOTO 1
      1700
11735 IF X9=2 THEN X3=X1-X2:GOTO 1
      1700
11740 IF X9=3 THEN IF X2=0 THEN 11
      775
11745 IF X9=3 THEN X3=X1/X2:GOTO 1
      1700
11750 IF X9=4 THEN X3=X1*X2:GOTO 1
      1700
11755 IF X9=5 THEN IF X1<0 THEN 11
      775
11760 IF X9=5 THEN X3=SQR(X1):GOTO
       11700
11765 IF X9=6 THEN X3=X1*X1:GOTO 1
      1700
11775 PRINT"OPERAZIONE IMPOSSIBILE
      ":FOR XX=1 TO 1500:NEXT:GOTO
       11700
11780 REM  VARIABILI: X9,X1,X2
11799 REM  NOME: MICROCALCOLATRICE
```

11800 Registra zona di memoria

(Solo per C-64)

A volte può sorgere la necessità di salvare una certa area di memoria (generalmente RAM) del calcolatore, sia che essa contenga semplici dati, oppure un programma in lin-

guaggio macchina. Sarebbe veramente uno scherzo effettuare l'operazione di salvataggio se il C-64 disponesse dell'istruzione BSAVE, comune in altre versioni basic.

Purtroppo non è così, e bisogna arrangiarsi in qualche maniera. Vi proponiamo perciò una routine che fa uso dei sottoprogrammi L.M. Kernal per la gestione dei file. Le subroutines usate sono:
SYS 65466: SETLFS, assegna i numeri logici del file e della periferica in uso (1 per registratore, 8 per drive).
SYS 65469: SETNAM, prepara i puntatori al nome del file.
SYS 65496: SAVE, operazione di salvataggio vera e propria.

Una trattazione più ampia dell'argomento, la potete trovare sul numero 20 di C.C.C. nell'apposito articolo dedicato, appunto, alle routine del Kernal.

Non è comunque necessario capire come funzionano le routine in Linguaggio Macchina (L.M.) per utilizzare il comodo sottoprogramma proposto. E' solo necessari o indicare i dati che il computer deve conoscere per effettuare il salvataggio corretto:

Alla domanda
NOME DEL FILE ?
è sempre consigliabile dare un qualsiasi nome se usate il lettore a cassette.

Per il disco invece è *obbligatorio:* se per distrazione inviate una stringa nulla, il programma non fa alcun salvataggio e ritorna con un messaggio d'errore.

Altre domande vengono poste:
INDIRIZZO INIZIALE ?
INDIRIZZO FINALE ?

Si riferiscono alle locazioni di memoria di inizio e di fine del file. L'indirizzo finale *deve* essere maggiore di quello inizale. Un apposito controllo elimina, ad ogni buon conto, la eventualità di un simile errore, altrimenti il computer andrebbe in confusione.

Alla domanda:
NASTRO O DISCO? (N/D)
premete il tasto corrispondente alla periferica di cui disponete.

• **Come gira il demo.** Nel demo di esempio (righe 100-250) sono stati assegnati, ad X3 e X4, gli indirizzi di inizio e fine della memoria video. In questo modo farete una copia esatta di tutto ciò che compare sullo schermo al momento del salvataggio.

Per effettuare una prova di funzionamento eseguite nell'ordine le seguenti operazioni tenendo conto che (R) significa: battere il tasto Return:

• Battete RUN (R)
• Quando compare il messaggio "Scrivi ciò che vuoi e, in seguito, digita RUN150", pasticciate con scritte varie lo schermo (anche con eventuali SYNTAX ERROR). Servirà per verificare che la schermata viene effettivamente registrata e, in seguito, caricata.
• Battete, quindi, RUN150
• Quando compare "Nome del file" digitate il nome che più vi aggrada e premete Return. Indicate in seguito, premendo "D" oppre "N", la periferica scelta. Seguite (se lavorate col nastro) i vari ordini impartiti dal C-64.
• Alla fine lo schermo viene cancellato e compare il messaggio "Vuoi caricare una schermata?" al quale dovete rispondere "S". Indicate il nome del File e premete Return. A questo punto il file viene caricato e visualizzato su schermo.
• Comparirà in reverse il messaggio "Per continuare premi un tasto" che eseguirete. La successiva visualizzazione del vettore A(I) dimostrerà che eventuali variabili definite prima del caricamento non sono state alterate.

Quando, infatti, viene caricato, da programma, un altro file-programma, è come se, al termine dell'operazione, venisse dato un GOTO alla prima riga del nuovo programma basic appena caricato. Poichè, nel nostro caso, il programma da caricare viene letto con la sintassi ",1", il programma basic "ufficiale" è ancora quello di prima. Fortunatamente, se l'ordine di caricare un programma viene dato sotto programma, le variabili precedentemente dichiarate non vengono cancellate. Ecco dunque la possibilità di utilizzare una variabile numerica (nel nostro caso HH) col significato di FLAG (tdeviatore) che faccia capire se il programma principale (il nostro DEMO) gira per la prima volta oppure no.

E' ovvio che, se si desidera memorizzare altre zone di memoria, nelle righe 160 e 170, al posto dell'istruzione PRINT è necessario assegnare il comando:

160 INPUT"DALLA LOCAZIONE";X3
170 INPUT"ALLA LOCAZIONE";X4

Per richiamare in memoria un file registrato in precedenza, battete semplicemente:

LOAD"NOMEFILE",8,1 per il disco
LOAD"NOMEFILE",1,1 per la cassetta.

F. Molinari

```
100 REM  ESEMPIO D'USO
110 REM  SALVA PORZIONE DI MEMOR
    IA
130 REM  SOLO PER C64
131 :
135 PRINTCHR$(18)"PER CONTINUARE
     PREMI UN TASTO"
136 GET A$:IF A$="" THEN 136
138 IF HH THEN 235
145 PRINT"SCRIVI CIO' CHE VUOI E
    , IN SEGUITO, DIGITA: RUN 15
    0":END
150 FOR I=1 TO 10:A(I)=I:NEXT
155 INPUT "NOME DEL FILE";X1$
```

```
  160 PRINT"INDIRIZZO INIZIALE 102
      4":X3=1024
  170 PRINT"INDIRIZZO FINALE 2023"
      :X4=2023
  180 PRINT"NASTRO O DISCO: (N/D)"
  190 GET X9$:IF X9$<>"N" AND X9$<
      >"D" THEN 190
  200 GOSUB 11800
  210 IF X0$="ERR" THEN PRINT"ERRO
      RE"
  220 PRINTCHR$(147)"VUOI CARICARE
       UNA SCHERMATA? (S/N)"
  225 GET A$:IF A$="" THEN 225
  230 IF A$="S" THEN INPUT "NOME D
      EL FILE";X1$:HH=1:LOAD X1$,8
      ,1
  235 PRINTCHR$(147)"ECCO IL PROSE
      GUIMENTO DEL PROGRAMMA"
  240 FOR I=1 TO 10:PRINTA(I):NEXT
  250 :
 9999 END
11800 X0$="":IF X3>=X4 OR (X1$=""
      AND X9$="D") THEN X0$="ERR":
      RETURN
11802 X1=1:IF X9$="D" THEN X1=8:X1
      $="0:"+X1$:CLOSE 15:OPEN 15,
      8,15,"X3"+X1$:CLOSE 15
11805 X2$=X1$:X2=PEEK(53)+256*PEEK
      (54)-LEN(X2$):POKE 782,X2/25
      6
11810 POKE 781,X2-PEEK(782)*256:PO
      KE 780,LEN(X2$):SYS65469:POK
      E 780,1:POKE 781,X1
11815 POKE 782,1:SYS65466:POKE 254
      ,X3/256:POKE 253,X3-PEEK(254
      )*256:POKE 780,253
11820 POKE 782,X4/256:POKE 781,X4-
      PEEK(782)*256:SYS65496
11825 IF (PEEK(783) AND 1) OR (ST
       AND 191) THEN 11835
11830 RETURN
11835 X0$"ERR":IF X1=1 THEN RETURN
11840 CLOSE 15:OPEN 15,8,15:INPUT#
      15,X3$,X4$:PRINTX3$;X4$:CLOS
      E 15:RETURN
11845 REM  VARIABILI: X1$,X3,X4,X9
      $,X0$,X1,X2,X3$,X4$
11850 REM  X1$: NOME FILE
11855 REM  X3,X4: INDIRIZZI INIZ E
       FIN
11899 REM  SAVE PORZIONI DI MEMORI
      A
```

11900 Scambia pagine video

(Solo per C64)

Come il nome lascia intuire, questa routine permette di gestire contemporaneamente due pagine di testo. Spieghiamo subito come utilizzarla.

La variabile X1 è l'unico parametro che il programma deve conoscere, e può assumere due valori:

X1=0 memorizza l'attuale pagina presente sul video cancellando irrimediabilmente l'altra.
X1=1 scambia fra di loro le due schermate, visualizzando quella memorizzata in precedenza.

A cosa può servire avere a disposizione due pagine di schermo?. Ad esempio nella presentazione di una maschera video. Una volta memorizzata, non bisogna più attendere che il computer la disegni tutte le volte che ne abbiamo bisogno. Una riga del tipo:

```
1000 X1=1:GOSUB 11900
```

renderà l'operazione praticamente istantanea. Questo grazie ad una routine che, manco a dirlo, è in linguaggio macchina (L.M.), e lavora su due buffer (=memoria temporanea), anzi tre, considerando anche lo schermo, di 1Kbyte ciascuno. Il primo è destinato alla memorizzazione; il secondo è utilizzato durante gli scambi come memoria temporanea di transito.

I due buffer (2048 byte in tutto), sono allocati nella RAM posta sotto la ROM del sistema operativo. Traduco, nella sostanza, il bisticcio di parole: in pratica non viene occupata memoria e non vi è alcun intralcio per i programmi basic.

Diamo un'occhiata alle POKE, abbastanza numerose, presenti nella routine:

POKE 679,4 numero di blocchi. Ogni pagina è composta infatti da 4*256=1024 byte. Sarebero sufficienti 1000 byte ma è più semplice realizzare un programma L.M. che ne memorizzi 1024.
POKE 251,0:POKE 252,4 byte basso e alto indirizzo sorgente.
POKE 253,0:POKE 254,160 byte basso e alto destinazione (le altre righe sono simili: cambiano solo gli indirizzi).

Le locazioni 251,252,253 e 254 sono dei puntatori in pagina zero che ci permettono di utilizzare (attenzione, si chiama proprio così): l'indirizzamento indicizzato indiretto.
Altra cacofonia, anche peggiore della precedente!

Questa tecnica (stiamo parlando di programmazione in L.M.), ci permette di trasferire porzioni di memoria in maniera molto efficente e compatta. Al solito, chi ne volesse sapere di

più, consiglio la lettura degli articoli e rubriche che C.C.C. regolarmente dedica all'argomento. In particolare il n. 19, dove viene dato ampio spazio proprio ai tipi di indirizzamento del processore 6502.

F. Molinari

```
  100 REM  ESEMPIO D'USO
  110 REM  MEMORIZZA E ALTERNA
  120 REM  2 PAGINE SCHERMO
  130 REM  SOLO PER C64
  140 :
  150 PRINTCHR$(147)CHR$(18)"PREMI
       UN TASTO"
  165 POKE 1024+RND(0)*1000,1:REM
       CREA PAGINA 1 CASUALMENTE
  170 GET XX$:IF XX$="" THEN 165
  180 PRINTCHR$(19)CHR$(18)"PAGINA
       N.1  ":X1=0:GOSUB 11900
  190 PRINTCHR$(147)CHR$(18)"PREMI
       UN TASTO"
  205 POKE 1024+RND(0)*1000,2:REM
       CREA PAGINA 2 CASUALMENTE
  210 GET XX$:IF XX$="" THEN 205
  220 PRINTCHR$(19)CHR$(18)"PAGINA
       N.2  "
  230 X1=1:GOSUB 11900:FOR I=1 TO
      1000:NEXT:GOTO 230
  240 :
 9999 END
11900 X9$="1201650010412541330011б
      0000174167002177251145253"
11905 X9$=X9$+"2002082492302522302
      542022082420881650010090011З
      3001096"
11910 X7=0:X8=0
11915 FOR X0=1 TO LEN(X9$) STEP 3:
      X9=VAL(MID$(X9$,X0,3))
11920 POKE 680+X8,X9:X8=X8+1:X7=X7
      +X9:NEXT
11925 IF X7<>4813 THEN PRINT"ERR X
      9$":END
11930 IF X1=1 THEN 11945
11935 POKE 679,4:POKE 251,0:POKE 2
      52,4:POKE 253,0:POKE 254,160
      :SYS680:RETURN
11945 POKE 679,4:POKE 251,0:POKE 2
      52,4:POKE 253,0:POKE 254,164
      :SYS680
11950 POKE 679,4:POKE 251,0:POKE 2
      52,160:POKE 253,0:POKE 254,4
      :SYS680
11955 POKE 679,4:POKE 251,0:POKE 2
      52,164:POKE 253,0:POKE 254,1
      60:SYS 680:RETURN
11991 REM  VARIABILI: X9$,X7,X8,X9
      ,X1
11992 REM  X1=0 MEMORIZZA PAGINA
11993 REM  X1=1 ALTERNA PAGINA
11999 REM  CSAMBIA PAGINE SCHERMO
```

12000 Koala-video

(Solo per C-64)

Il KoalaPainter, con tavoletta grafica o meno, è certamente uno dei più usati e conosciuti programmi per disegnare in alta risoluzione.

Con un po' di fantasia (e sopratutto pazienza), si possono creare disegni multicolori veramente belli, e salvarli o caricarli da dischetto. Qualcuno però si sarà chiesto: "Bello, ma cosa me ne faccio se non posso utilizzarli all'interno dei *miei* programmi, dato che per caricarli devo avere in memoria il Koala?"

Risposta: copiate questa routine e finalmente i *vostri* capolavori saranno lì pronti per l'uso: Scenario di un gioco, un variopinto menù eccetera.

Il programma contiene una breve routine in linguaggio macchina, necessaria per spostare la pagina video e la memoria colore nelle giuste posizioni (operazione che in basic richiederebbe troppo tempo):
24576: inizio pagina grafica (8Kbyte)
23552: inizio video (1Kbyte)
55296: inizio memoria colore (1Kbyte)

Infine, con istruzioni POKE e PEEK, viene visualizzata la pagina grafica che, per il Koala, è del tipo multicolore.

Un'avvertenza: nella variabile X1$ dovete memorizzare il nome *esatto* del file registrato, compreso il simbolo grafico CHR$(129) all'inizio della stringa (ottenibile anche con la pressione dei tasti CBM e 1) e gli spazi vuoti; pena la comparsa del messaggio FILE NOT FOUND. Ma queste ultime informazioni dovrebbero esser superflue per chi possiede il Koala.

F. Molinari

```
100 REM  ESEMPIO D'USO
105 REM  SOLO PER C-64
110 REM  CARICARE UN
115 REM  DISEGNO ESEGUITO
120 REM  CON IL KOALA PAINTER
125 :
130 IF XX THEN 137
131 PRINTCHR$(147)"INSERISCI IL
    DISCO CON UN DISEGNO KOALA E
     PREMI UN TASTO"
132 GET A$:IF A$="" THEN 132
```

```
  134 X1$=CHR$(129)+"PIC*":PRINT"A
      TTENDERE..."
  135 IF XX=0 THEN XX=1:LOAD X1$,8
      ,1
  137 PRINT"DOPO AVER OSSERVATO IL
       DISEGNO"
  140 PRINT"PER RITORNARE IN MODO
      TESTO PREMERE IL TASTO <RETU
      RN>"
  150 PRINT"PREMI UN TASTO"
  160 GET XX$:IF XX$="" THEN 160
  170 GOSUB 12000
  180 GET XX$:IF XX$<>CHR$(13) THE
      N 180
  190 POKE 56576,151:POKE 53272,21
  200 POKE 53270,PEEK(53270) AND 2
      39
  210 POKE 53265,PEEK(53265) AND 2
      23
  215 PRINTCHR$(147)
  220 :
 9999 END
12000 X9$="16000017416700217725114
      525320020824923025223025420
      208242096"
12002 X7=0:X8=0
12005 FOR X0=1 TO LEN(X9$) STEP 3:
      X9=VAL(MID$(X9$,X0,3))
12010 POKE 680+X8,X9:X8=X8+1:X7=X7
      +X9:NEXT
12015 IF X7<>3700 THEN PRINT"ERROR
      E":END
12020 POKE  679,4:POKE  251,64:POK
      E  252,127:POKE  253,0:POKE
       254,92:SYS 680
12025 POKE  679,4:POKE  251,40:POK
      E  252,131:POKE  253,0:POKE
       254,216:SYS 680
12030 POKE 56576,150
12035 POKE 53272,126
12040 POKE 53270,PEEK(53270) OR 16
12045 POKE 53265,PEEK(53265) OR 32
12050 RETURN
12055 :
12056 REM  VARIABILI: X9$,X7,X8,X0
      ,X9
12060 REM  VISUALIZZA LA PAGINA GR
      AFICA
12065 REM  DISEGNATA CON IL KOALAP
      AINTER
12099 REM  NOME: KOALAPRINTER (SU
      VIDEO)
```

Elenco delle routine pubblicate

Qui di seguito viene riportato l'indice di tutte le routine finora pubblicate (fra parentesi il numero di Commodore Computer Club su cui si possono trovare).

Software Club

Le riviste su cassetta della Systems Editoriale si fondono per prepararsi ad un'importante sforzo di diffusione internazionale del nostro software.

Sino ad ora abbiamo offerto ai nostri utenti due diverse testate su supporto magnetico: Commodore Club e 16/48.

La prima era evidentemente destinata agli utenti Commodore, mentre la seconda era diretta a tutti coloro che possedessero un computer Sinclair.

La nuova rivista su cassetta, denominata "Software Club", è destinata anche alle persone che possiedono un calcolatore MSX.

Questa nuova testata sarà la prima ad uscire contemporaneamente nei più importanti paesi europei.

Poichè le tre marche, Commodore, Sinclair e MSX, rappresentano complessivamente la porzione più importante del mercato europeo degli home computer, abbiamo ritenuto opportuno fornire un'unico prodotto che fosse in grado di accontentare contemporaneamente le tre diverse fasce di utenti.

Commodore Club consegna idealmente il testimone a questa nuova pubblicazione su supporto magnetico, nella speranza di fornire un prodotto sempre più adeguato alle esigenze di ogni singolo utente.

Questa nuova veste editoriale servirà ad esaltare ulteriormente, la qualità del software che vi proporremo nel corso dei vari mesi.

Ringraziandovi anticipatamente dell'interesse di cui sarà oggetto auesta nuova rivista su cassetta, vi rimandiamo al primo appuntamento in edicola con questa pubblicazione.

La redazione di Software Club

FINALMENTE SOFTWARE CLUB

Nel primo numero di Software Club ogni utente di un modello Commodore (C.64, C.20 e C.16) potrà trovare degli argomenti di suo interesse.

Per quanto concerne il 64, vi proponiamo una nuova spettacolare Adventure di stile tipicamente orientale.

Le qualità richieste per eccellere in questo gioco sono:
intuito, senso dell'orientamento e capacità di interpretazione delle immagini.

Parecchie delle persone che hanno acquistato un Commodore 64, sono particolarmente interessate alle sue applicazioni nel campo della grafica.

Già nel passato la rivista su carta Commodore Computer Club si è occupata di questo problema, fornendo un tool di istruzioni grafiche.

La versione che vi proponiamo è molto più interessante rispetto a quella precedentemente offerta, poichè contempla al suo interno parecchi nuovi comandi che prima mancavano.

Anche sul Commmodore 64 è possibile realizzare un sequenza filmica attraverso un attento lavoro di programmazione.

Se siete curiosi di osservarla non mancate a questo primo appuntamento.

Naturalmente anche gli appassionati di video game rimarranno soddisfatti dallo spazio dedicato ai giochi di azione.

Dopo esserci occupati dei più importanti personaggi della narrativa infantile, concludiamo questo ciclo con Cappuccetto Rosso.

E' un elettrizzante gioco di azione, interamente realizzato in linguaggio macchina, che richiede prontezza e abilità nel manovrare il joystick.

Questi sono solo alcuni dei temi più interessanti di questa pubblicazione, che racchiude complessivamente ben 21 programmi.

Rammentiamo a tutti coloro che possiedono un computer Commodore 16 che abbiamo realizzato per loro un potente programma di archivio dati: Data-base

Ora analizzeremo gli argomenti che verranno trattati nei numeri speciali della nostra pubblicazione su cassetta Software Club.

Dynastopoli

Questo numero speciale della cassetta è dedicato agli utenti dei computer C.64, C.20, C.16.

Per quanto concerne il C.64 potrete trovare degli interessanti giochi di simulazione (Dynastopoli, gioco che da il nome alla cassetta), d'azione (castelli, bigio, uomini volanti) e programmi di utilità.

Questa interessante cassetta che non dovrà assolutamente sfuggirvi.

Gestione domestica

Nel numero passato di questa rivista, abbiamo descritto particolareggiatamente i programmi destinati alla gestione domestica.

Questo speciale vi consentirà di gestire, in maniera accurata, tutti i più diversi aspetti della vostra vita familiare (bilancio, dieta, scheda medica).

Velocissimo Basic

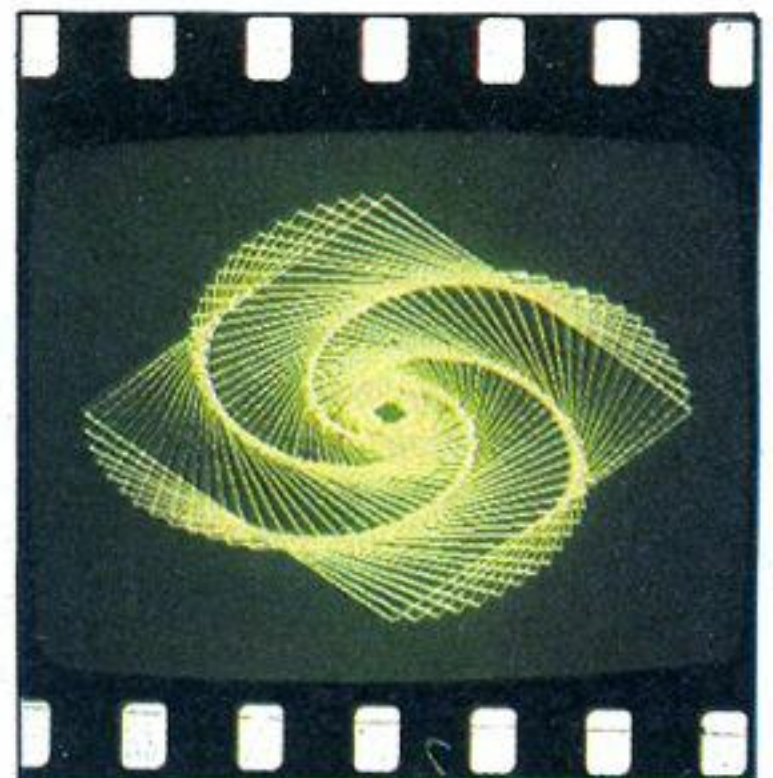
Chi si avvicina per la prima volta al mondo dei computer, può incontrare delle difficoltà nella comprensione del linguaggio che occorre utilizzare per dialogare con queste macchine.

Questo corso, articolato in 4 cassette e destinato agli utenti dei computer Commodore 64, Spectrum ed MSX, vi guiderà passo per passo all' aquisizione di questo nuova lingua: il Basic.

Una delle novità più importanti di questa iniziativa consiste nel fatto che non necessità di alcun supporto di manualistica.

Anche coloro che abbiano già confidenza con questo linguaggio, potranno trovare degli interessanti spunti di approfondimento all'interno di ciascuna lezione.

La nuova voce

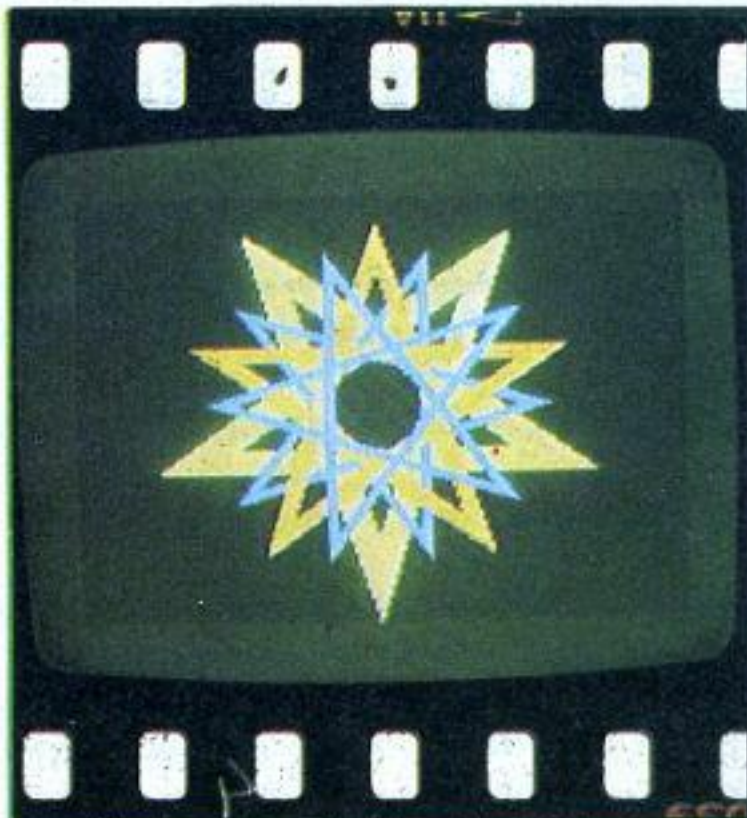
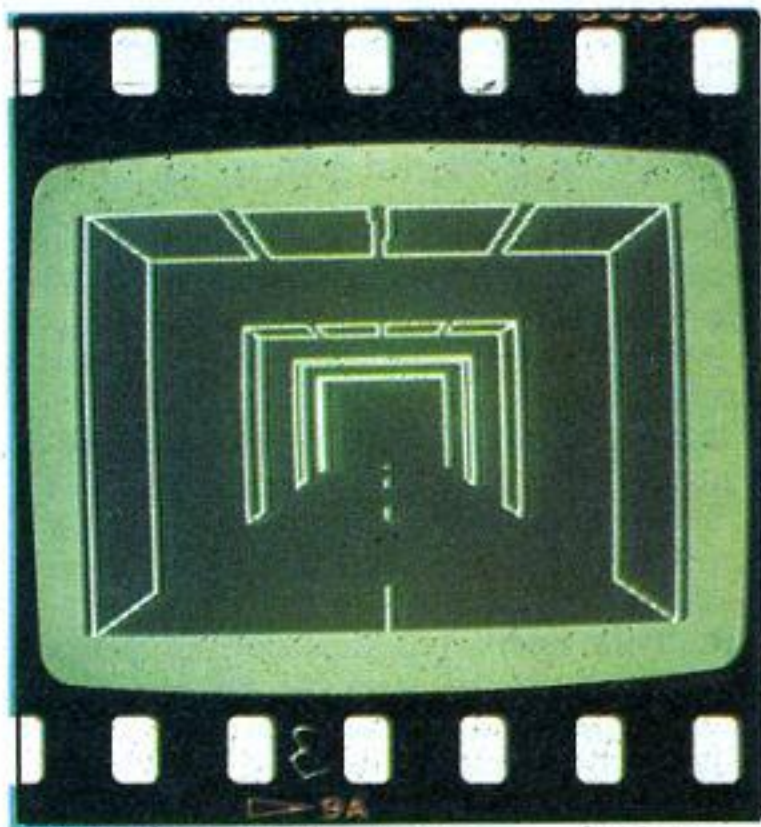
Quando l'anno scorso proponemmo ai nostri lettori il programma la voce, che consentiva al computer di riprodurre il nostro modo corrente di dialogare, ottenemmo un notevole gradimento presso tutte le nostre diverse fasce di utenti.

Memori di questa eperienza, vi offriamo un nuovo prodotto più avanzato e coerente con l'esigenze dei nostri lettori.

Nello speciale troverete un nuovo comando basic, voce, che vi consentirà di introdurre dei commenti parlati all'interno dei programmi di vostra creazione.

Inoltre vi proponiamo la prima adventure parlata in italiano, dove i commenti renderanno ancora più avvincente la vostra competizione con il computer.

Non poteva naturalmente mancare un video-game che utilizzasse il programma di sintesi vocale ciò accentuerà il mordente delle diverse fasi di gioco.

In questa cassetta sono contenute anche delle applicazioni inerenti al tema della didattica.

Rammenterete senza dubbio il famoso grillo parlante: anche sul vostro Commodore 64 potrete ottenere i medesimi effetti mediante questo programma.

Speciale musica

Il Commodore 64 è l'unico dei computer appartenenti alla fascia dell'hobbismo ad essere in possesso di un generatore sonoro particolarmente sofisticato: il SID.

Il tema della musica è uno degli argomenti più cari alla maggior parte degli utenti di questa macchina e abbiamo ritenuto necessario dedicargli un numero speciale della nostra pubblicazione.

In questa cassetta sono contemplati dei sofisticati programmi che consentono di generare musica con il Commodore 64.

Inoltre vi proponiamo un'enorme varietà di brani già composti, che potranno essere semplicemente ascoltati oppure utilizzati, come motivo ricorrente, all'interno di un vostro programma.

Gli autori, di alcuni di essi, sono:

Musica moderna

Madonna-Police-Duran Duran-Band Aid-Usa for Africa-Sandy Marton-Spandau Ballet-Tina Turner-Style Council-Bruce Springsteen-King-Jagger e Bowie

Anni 60/70

Beatles-Led Zeppelin-Simon and Garfunkel Rolling Stones-Pink Floyd-Genesis

Italiani

Battisti-Paoli-Mina-Baglioni-Dalla-Vasco Rossi-Pooh-Vanoni-De Andre-Dario Baldan Bembo

Tutto il lavoro di creazione di questo speciale è stato coordinato dal famoso cantante e autore Dario Baldan Bembo.

Oroscopo 64

Anche un computer può contribuire ad alimentare l'atmosfera natalizia.

In questa cassetta, assieme a degli interessanti giochi di società, troverete un precisissimo programma che vi fornirà le previsioni astrali per tutto il prossimo anno.

Buon divertimento.

Speciale utility

Molte volte vi sarà capitato, utilizzando un Commodore 64, di aver bisogno di un programma che vi agevolasse nello svolgere determinate operazioni.

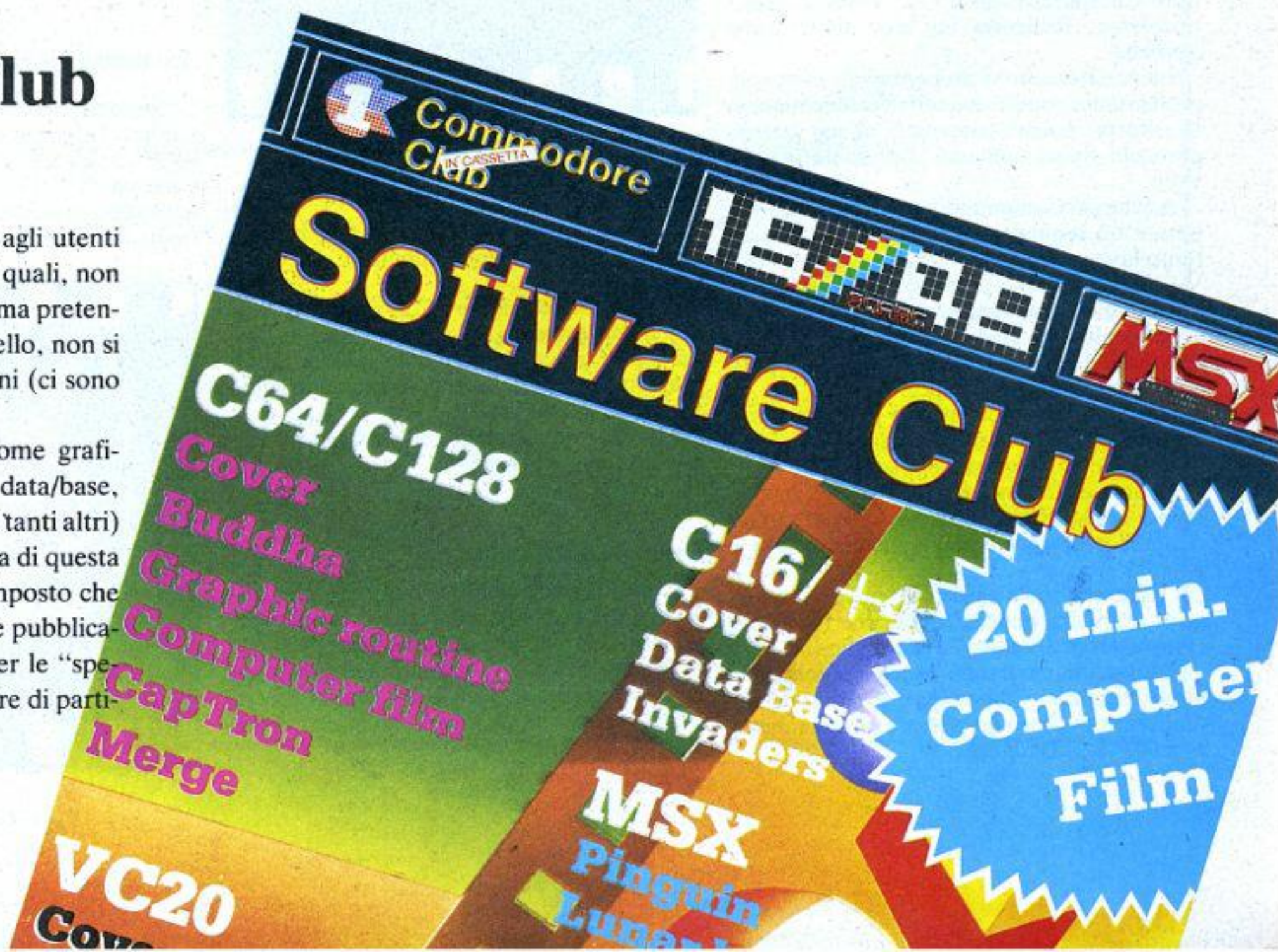
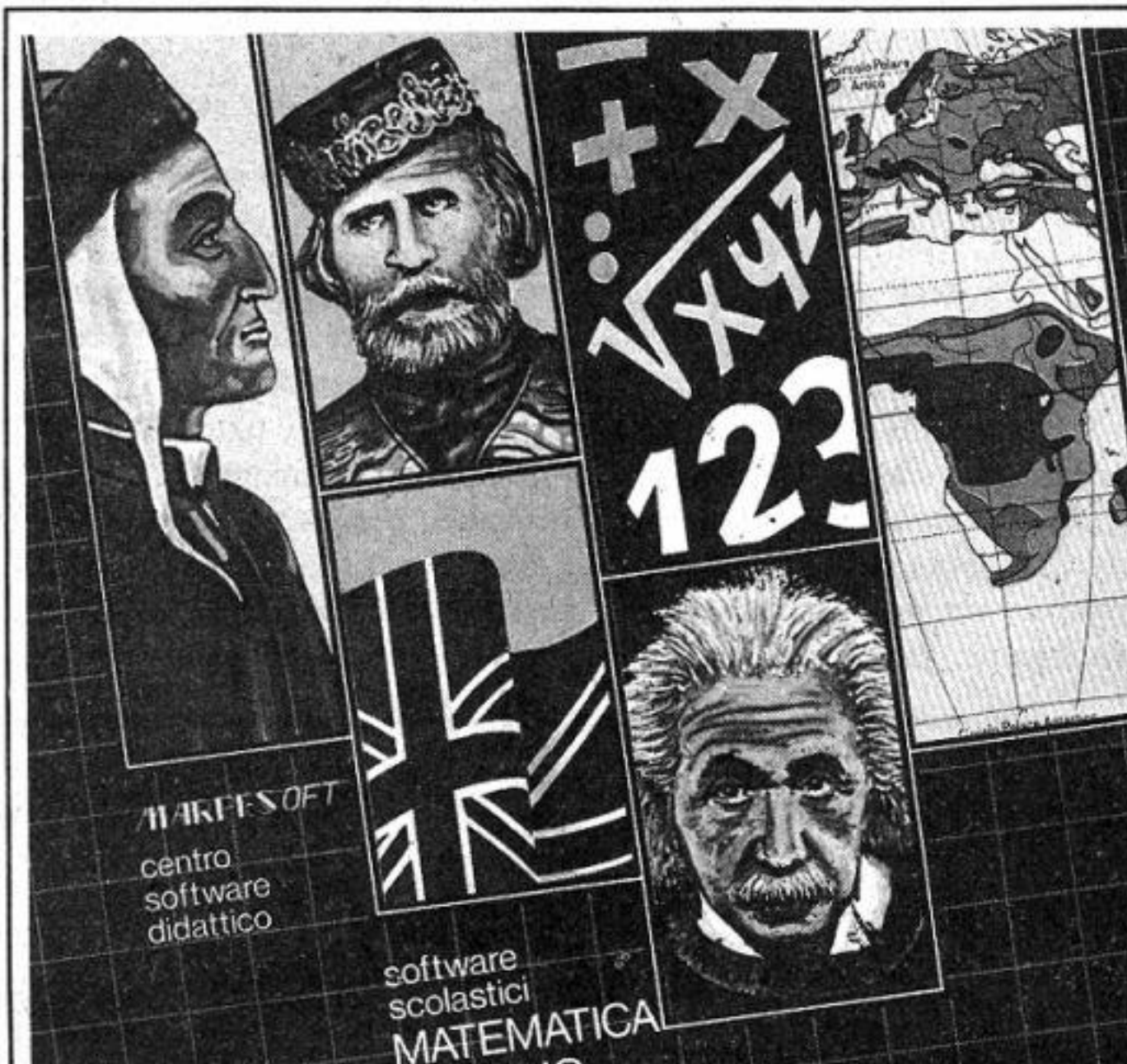
Tramite questa cassetta potrete usufruire di potenti programmi che renderanno di parecchie volte più veloce il caricamento da registratore o da disco.

Inoltre avrete la possibilità di usufruire di un editor che vi consentirà di lavorare su 80 colonne durante ogni fase di utilizzo del vostro computer.

Queste sono solo alcune delle utility che vi proponiamo in questo numero speciale della nostra pubblicazione, mentre tutte le altre rappresenteranno una piacevole sorpresa per ciascuno di voi.

Software Club

Rivista su nastro destinata agli utenti Commodore, Sinclair, MSX i quali, non volendo digitare lunghi listati ma pretendendo sofware di un certo livello, non si accontentano dei soliti giochini (ci sono però anche quelli...)

Applicazioni particolari (come grafica, musica, word/processor, data/base, spreadhseet, mini gestionale e tanti altri) rappresentano il punto di forza di questa rivista su cassetta al prezzo imposto che sfiora il ridicolo: L.7000 per le pubblicazioni "normali", e L.12000 per le "speciali" contenenti, cioè, software di particolare pregio.

Fare i compiti col C-64

Alessandro de Simone

Può il tuo computer aiutarti nei lavori scolastici da svolgere a casa? I prodotti della Marpesoft sembrerebbero dire di sì....

La Marpesoft Centro Software Didattico si presenta sul mercato con un gruppo di prodotti destinati ad applicazioni scolastiche.

I "volumi" da noi provati si riferiscono a cassette, da caricare col solito registratore, che trattano diverse discipline: Inglese, Italiano, Fisica, Matematica, Geografia, Storia.

La struttura di ogni programma è sempre la stessa: caricato il programma (che parte automaticamente al termine dell'operazione) compare un menu con un certo numero di scelte tra cui, ovviamente, le istruzioni per un corretto utilizzo del programma.

La metodologia utilizzata è quella del questionario a scelta multipla, vale a dire una domanda - esercizio seguita da tre risposte possibili contraddistinte ciascuna da un numero. Premendo uno dei tasti idonei per indicare la risposta ritenuta esatta, compare il messaggio esatto oppure errato seguito, o meno, dall'indicazione della risposta corretta.

E' infatti possibile, all'inizio, richiedere che la risposta esatta, in caso di errore, venga evidenziata oppure no, come pure è possibile "passare" alla successiva domanda rinviandola alla fine dell'esercizio.

E' inoltre facoltà dello studente fissare il tempo massimo di 15 secondi entro cui assegnare la risposta.

Le domande per ciascuno dei sette "capitoli" sono 15 e non sempre semplicissime.

Indichiamo di seguito ciò che compare in una videata del corso di Inglese in modo da rendere l'idea di come è strutturato il programma:

WHEN HE IS ... HE GOES TO BED

1 RIGHT
2 THIRSTY
3 SLEEPY

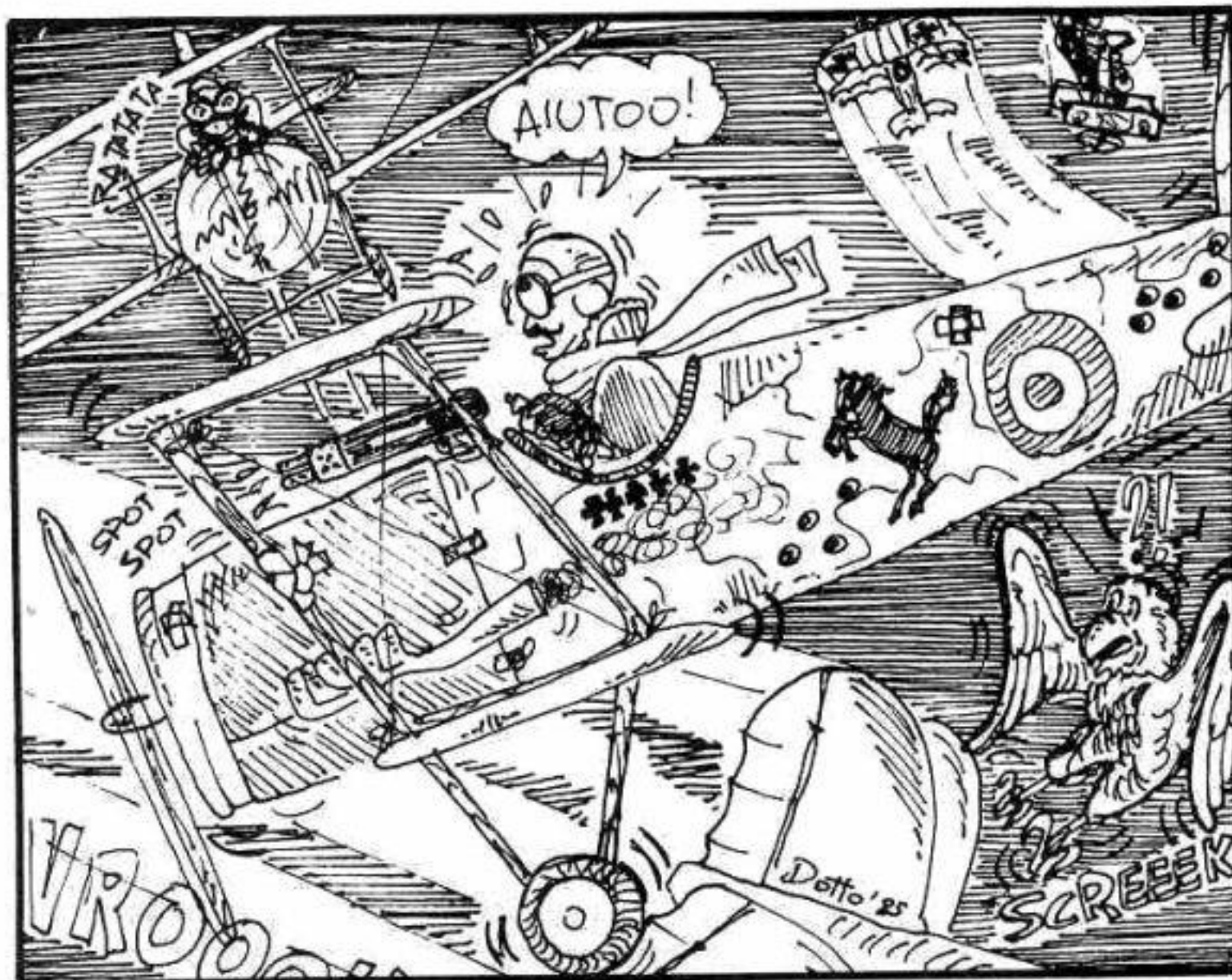
Qual'è la tua risposta?

La risposta esatta, per chi conosce l'Inglese, è la terza ed è quindi necessario premere il tasto "3" per ottenere in risposta il messaggio "ESATTO" ed il passaggio alla domanda successiva.

Alla fine dell'esercitazione viene indicato il numero delle risposte esatte, quello delle errate ed il tempo trascorso.

Nonostante, come era intuibile, i programmi esaminati non si sostituiscano ai libri di testo nè tantomeno svolgano i compiti assegnati a casa dai professori, purtuttavia possono rivelarsi utili come "ripasso" di argomenti precedentemente affrontati a scuola.

Attacco aereo

di Carlo *e* Lorenzo Barazzetta

Questo gioco, in parte in BASIC e in parte in linguaggio macchina, consente anche ai lettori inesperti di apportare modifiche grazie alle numerose REM che indicano le POKE da alterare.

Risulta quindi semplice cambiare i colori degli aerei, i loro ritardi negli spostamenti (e relative difficoltà di manovra), i volumi sonori degli effetti "speciali" eccetera.

Mediante i tasti "B" ed "N", guidate un aereo che deve sfuggire alle collisioni con altri aerei (sprite) che sopraggiungono. Potete, a vostra volta, posizionarvi alle loro spalle e sparare col tasto Shift. Immediatamente partirà un proiettile che, raggiunto l'aereo avversario, gli cambierà colore. Se, invece, urtate contro gli aerei, esplodete ed il gioco ricomincia premendo il tasto F7.

Simpatico il rumore dei jet, davvero realistico, e nuova è l'animazione dell'aereo che, nello spostarsi a destra e a sinistra, si inclina come se fosse uno "vero".

```
100 REM  COMMODORE 64
110 REM  ATTACCO AEREO
120 :
130 SYS65409:REM  AZZERA SPITE
140 H=2040:P=16128:K=36864
150 V=53248:SI=54272
160 FOR A=0 TO 24:POKE SI+A,0:NE
    XT
170 POKE V+32,0:REM  COLORE BORD
    O
180 POKE V+33,6:REM  COLORE SFON
    DO
190 POKE 646,PEEK(V+32) AND 15
200 PRINT:PRINT"ATTACCO AEREO (A
    TTENDERE 127 E 327)":PRINT
210 PRINT"SCRITTO DA CARLO"CHR$(
    32);
220 PRINT"E LORENZO BARAZZETTA":
    PRINT
230 PRINT"USA I TASTI <B> E"CHR$
    (32);
240 PRINT"<N> PER SPOSTARTI":PRI
    NT
250 PRINT"USA IL TASTO <SHIFT>";
260 PRINTCHR$(32)"PER SPARARE":P
    RINT
270 FOR A=0 TO 127:READ B:PRINTC
    HR$(19)A
280 IF B>=0 AND B<256 THEN 300
290 PRINT"ERRORE NEL BLOCCO 1":E
    ND
300 C=C+B:POKE P+64+A,B:NEXT
310 IF C<>8494 THEN 290
320 FOR A=0 TO 63:POKE P+A,0:NEX
    T
330 FOR A=1 TO 24 STEP 3:POKE P+
```

```
    A,16:NEXT
340 C=0:FOR A=0 TO 327:READ B:PR
    INTCHR$(19) TAB(10)A
350 IF B>=0 AND B<256 THEN 370
360 PRINT"ERRORE NEL BLOCCO 2":E
    ND
370 C=C+B:POKE K+A,B:NEXT
380 IF C<>38388 THEN 360
390 FOR A=0 TO 20:FOR B=-1 TO 1
400 POKE 2,PEEK(P+65+B+A*3)
410 SYSK:POKE (P+193-B+A*3),PEEK
    (780)
420 NEXTB,A:SYSK+9
430 POKE K+21,16:REM  RITARDO GE
    NERALE
440 POKE K+31,40:REM  RIT.SPOSTA
    MENTO
450 POKE K+38,28:REM  TASTO SINI
    STRA
460 POKE K+46,39:REM  TASTO DEST
    RA
470 POKE K+79,12:REM  RITARDO PR
    OIETTILE
480 POKE K+240,80:REM  RIT.SPOST
    .AEREI
490 POKE K+330,32:REM  RITARDO A
    EREO 1
500 POKE K+331,40:REM  RITARDO A
    EREO 2
510 POKE K+332,48:REM  RITARDO A
    EREO 3
520 POKE K+333,56:REM  RITARDO A
    EREO 4
530 POKE K+334,64:REM  RITARDO A
    EREO 5
540 POKE K+335,72:REM  RITARDO A
    EREO 6
550 PRINTCHR$(147);:POKE V+22,17
560 FOR A=1 TO 25:PRINT
570 PRINTSPC(28)CHR$(18);:FOR B=
    1 TO 11
580 PRINTCHR$(32);:NEXTB,A
590 PRINTCHR$(19)
600 PRINTSPC(31)CHR$(18)"PUNTI"
610 PRINTSPC(31)CHR$(18)"00000"
620 POKE V,128:POKE V+1,140
630 POKE V+39,0:REM  COLORE AERE
    O
640 POKE V+46,1:REM  COLORE PROI
    ETTILE
650 FOR A=2 TO 15:POKE V+A,1:NEX
    T
660 POKE V+27,255
670 FOR A=0 TO 6:POKE H+A,254:NE
    XT
680 POKE H+7,252
690 FOR A=0 TO 10:POKE 32768+A,1
    :NEXT
700 A=12:REM  VOLUME MOTORE
710 POKE SI+5,0:POKE SI+6,A*16
720 A=12:REM  VOLUME PROIETTILE
730 POKE SI+12,0:POKE SI+13,A*16
740 POKE SI+14,255:POKE SI+15,25
    5
750 A=8:REM  VOLUME
760 POKE SI+24,A OR 128:POKE SI+
    18,129
770 POKE SI+4,129
780 POKE V+21,127:POKE V+30,0
790 SYSK+19
800 POKE SI+4,128:POKE SI+11,32
810 A=15:REM  VOLUME SCOPPIO
820 POKE SI+5,15:POKE SI+6,A*16+
    8
830 POKE SI+1,1:POKE SI+4,129
840 FOR A=1 TO 100:POKE H,RND(1)
    *16+64
850 POKE V+39,RND(1)*16
860 NEXT:POKE SI+4,128
870 POKE V+21,PEEK(V+21) AND 254
880 POKE 198,0:PRINT
890 PRINTSPC(30)CHR$(18)"PREMI F
    7";
900 A=3:REM  TASTO PER RIPARTIRE
910 IF PEEK(203)<>A THEN 910
920 FOR A=1 TO 8:PRINTCHR$(157);
    :NEXT
930 FOR A=1 TO 8:PRINTCHR$(32);:
    NEXT
940 GOTO 590
950 REM  INIZIO DATI BLOCCO 1
960 DATA  0,16,0,0,16,0,0,56,0,0
    ,56,0
970 DATA  0,56,0,0,56,0,0,92,0,0
    ,20,0
980 DATA  0,150,0,0,150,192,1,24
    7,64
990 DATA  1,247,64,3,251,128,3,2
    51,128
1000 DATA  7,125,192,7,125,192,6,
    56,192
```

```
1010 DATA  0,40,0,0,76,0,0,142,0,
     0,142,0
1020 DATA  0,0,16,0,0,16,0,0,56,0
     ,0,56,0
1030 DATA  0,56,0,0,124,0,0,238,0
     ,1,69,0
1040 DATA  3,69,128,27,69,176,23,
     125
1050 DATA  208,31,125,240,63,255,
     248
1060 DATA  126,124,252,252,124,12
     6,248
1070 DATA  124,62,240,56,30,0,40,
     0,0
1080 DATA  238,0,1,239,0,1,239,0,
     0
1090 REM  FINE DATI BLOCCO 1
1100 REM  INIZIO DATI BLOCCO 2
1110 DATA  162,7,6,2,106,202,16,2
     50,96
1120 DATA  24,162,62,62,192,63,20
     2,16
1130 DATA  250,96,216,160,16,136,
     208
1140 DATA  253,206,0,128,208,43,1
     69,40
1150 DATA  141,0,128,165,203,201,
     28,208
1160 DATA  4,169,253,208,10,201,3
     9,208
1170 DATA  4,169,255,208,2,169,25
     4,141
1180 DATA  248,7,56,233,254,24,10
     9,0
1190 DATA  208,141,0,208,74,74,14
     1,1
1200 DATA  212,206,1,128,208,117,
     169
1210 DATA  12,141,1,128,173,21,20
     8,41
1220 DATA  128,208,31,173,141,2,4
     1,1
1230 DATA  240,51,169,132,141,15,
     208
1240 DATA  173,0,208,141,14,208,1
     73,21
1250 DATA  208,9,128,141,21,208,1
     69,33
1260 DATA  141,11,212,173,15,208,
     73,255
1270 DATA  10,141,8,212,206,15,20
     8,208
1280 DATA  13,173,21,208,41,127,1
     41,21
1290 DATA  208,169,32,141,11,212,
     173
1300 DATA  30,208,74,144,1,96,10,
     10,144
1310 DATA  36,141,2,128,162,5,14,
     2,128
1320 DATA  144,3,32,27,145,202,16
     ,245
1330 DATA  162,4,254,111,4,189,11
     1,4
1340 DATA  201,186,208,8,169,176,
     157
1350 DATA  111,4,202,16,238,162,5
     ,222
1360 DATA  3,128,208,29,189,74,14
     5,157
1370 DATA  3,128,138,72,10,170,22
     2,3
1380 DATA  208,208,12,173,27,212,
     157
1390 DATA  2,208,104,72,170,32,27
     ,145
1400 DATA  104,170,202,16,219,206
     ,9,128
1410 DATA  208,41,169,80,141,9,12
     8,162
1420 DATA  5,189,249,7,56,233,254
     ,168
1430 DATA  138,72,10,170,152,24,1
     25,2
1440 DATA  208,157,2,208,104,170,
     173
1450 DATA  27,212,41,63,208,3,32,
     48,145
1460 DATA  202,16,222,76,20,144,1
     73,33
1470 DATA  208,41,15,141,10,128,1
     73,27
1480 DATA  212,41,15,205,10,128,2
     40,238
1490 DATA  157,40,208,173,27,212,
     48,4
1500 DATA  169,254,208,11,173,27,
     212
1510 DATA  48,4,169,253,208,2,169
     ,255
1520 DATA  157,249,7,96,-1
1530 REM  FINE DATI BLOCCO 2
```

Prossimamente sui vostri schermi

Per il Commodore 64 è stato probabilmente sviluppato il più ampio parco di programmi esistente per home computer, tra cui moltissimi giochi. Vogliamo presentarne due, opera di due software house altrettanto "grandi": la Epyx e la Elite.

di **Giancarlo Mariani**

Winter Games (Epyx)

Molti di voi, circa un anno e mezzo fa, saranno rimasti meravigliati di fronte a *Summer Games* (giochi estivi), che la Epyx proponeva, appunto, agli inizi dell'estate 1984.

Questo gioco, è composto da numerose gare olimpiche, da disputarsi contro il computer o contro altri avversari. Summer Games era (ed è tuttora) notevole dal punto di vista grafico, delle animazioni, per gli effetti sonori, eccetera. Probabilmente nessuno, dopo averlo visto, avrà pensato che si potesse sviluppare qualcosa di meglio.

La Epyx ha riproposto, un anno dopo, *Summer Games II,* che ha lasciato letteralmente a bocca aperta gran parte degli utilizzatori del C-64 (ma anche di altri computer).

Questo gioco, infatti, era stato (sembra impossibile) addirittura migliorato in grafica, suoni, velocità, movimenti, rispetto al primo, ottenendo un prodotto eccezionale sotto ogni punto di vista.

Considerato quindi il (meritato) successo dei due programmi anzidetti, ecco che la Epyx suggerisce un nuovo programma che non è (come qualcuno potrebbe malignamente pensare) Summer Games III, bensì *Winter Games* (Giochi invernali).

Il programma, come i precedenti, è basato su attività sportive che, si disputano sulla neve.

La grafica, i vari effetti e i suoni sono addirittura migliorati.

Dopo aver caricato il programma, vi sono le "solite" cerimonie di apertura con sottofondo musicale. Stavolta però tutto il campo è ricoperto di neve. Dopo la presentazione (interrompibile anche prima che giunga a termine) il programma presenta un menu con sette opzioni selezionabili dal joystick posto indifferentemente in porta 1 oppure 2.

Da menu è possibile: gareggiare in tutti (o solo in alcuni) sports, in un gioco soltanto, fare pratica in una gara, scegliere il numero dei joysticks, esaminare i record mondiali e rivedere le cerimonie di apertura.

Se l'opzione scelta è una delle prime tre, viene chiesto il nome dei giocatori (fino a un massimo di otto) e nel caso ve ne sia solo uno, questi avrà come antagonista il computer.

Dopo aver chiesto il nome, il C64 chiede anche la nazionalità del giocatore. Muovendosi con il joystick ci si deve posizionare sulla bandiera del Paese desiderato (in tutto sono 18, Italia compresa) e premere il pulsante in modo da ascoltare l'inno nazionale del Paese scelto.

Dopo aver introdotto il nome e la nazionalità dei vari giocatori ed aver dato la conferma della correttezza dei dati inseriti, un altro menu comprende i giochi selezionabili, in tutto sette.

1 - Hot Dog:

E' il salto acrobatico con sci. Dopo essere partiti si deve muovere il joystick

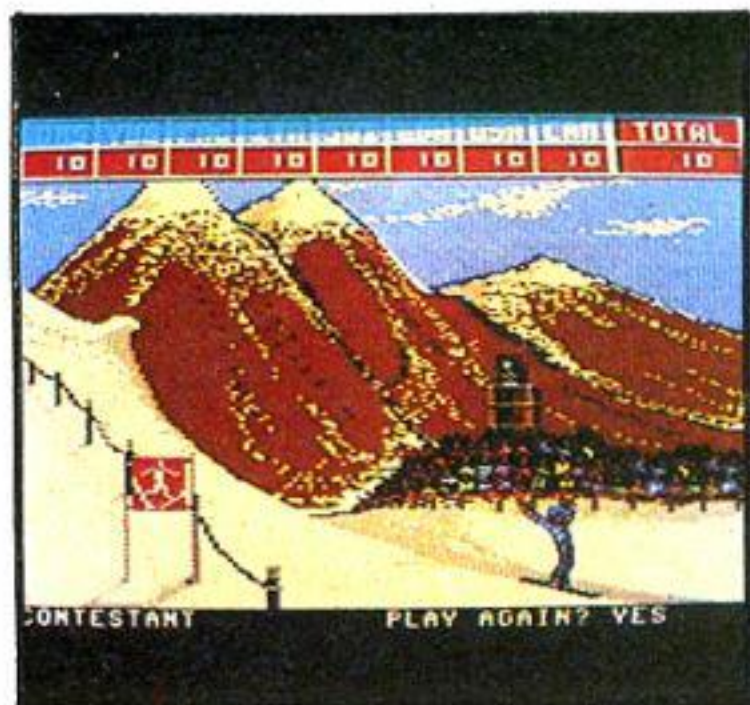
per cercare di fare acrobazie: spostando la leva a destra o a sinistra si faranno rispettivamente capriole in avanti o all'indietro. Spostandola in alto o in basso si avrà una stabilizzazione dello sciatore, mentre con posizioni intermedie si avranno altri effetti, quali alzata di gambe, spostamento in avanti/indietro, eccetera.

Il punteggio è determinato dal numero e dalla qualità delle acrobazie svolte nel breve tempo che si ha a disposizione. Bisogna stare attenti a non sbilanciarsi troppo, altrimenti si cade e si perdono punti. Questi (come per altre gare) vengono assegnati da otto giudici di gara che ne hanno a disposizione dieci ciascuno. Vi assicuriamo, però, che è veramente difficile ottenere il massimo punteggio.

Una splendida musica di sottofondo vi accompagna per tutto il gioco che, dotato di grafica e velocità di movimento eccezionali, conferma l'alta qualità del prodotto.

2.- Biathlon

E' una gara di percorso sugli sci, composta da quattro fasi, che viene svolta individualmente da ciascun partecipante.

Per prima cosa si dovrà oltrepassare un ponte in pianura, sciando il più velocemente possibile. In seguito vi sarà una rilassante discesa, in preparazione della salita, lungo la quale dovremo consumare tutte le nostre forze (ed anche il joystick...) per superarla. Compiuto per la terza volta lo stesso percorso, ci sarà un momento di pausa durante il quale, con un fucile in continuo movimento, dovremo colpire cinque bersagli fissi.

Il tutto si ripete quattro volte prima che venga assegnato il punteggio, che dipende non solo dal tempo impiegato ma anche dal numero di bersagli colpiti.

3 - Figure Skating

Tra le gare invernali non poteva certo mancare il pattinaggio artistico.

A costo di ripetere cose già dette, si sottolinea volentieri il nostro apprezzamento per la grafica e l'animazione della ragazza che pattina: sembra di vedere un cartone animato.

Anche questo sport, come il precedente, deve essere giocato individualmente. Nel tempo di un minuto, dovremo, attenendoci alle regole del pattinaggio artistico (e, ovviamente, muovendo il joystick) eseguire acrobazie, tra cui giravolte, salti, eccetera senza, ovviamente, cadere.

Il punteggio è dato dalla correttezza degli esercizi svolti.

La gara è accompagnata da un'allegra musichetta, che segue il "ballo" della ragazza.

4 - Ski jump

E' il classico salto dal trampolino. Dopo averlo spiccato, in alto a destra compare un riquadro, indicante l'assetto del giocatore durante il "volo", che può essere modificato mediante il joystick, cercando di raggiungere una corretta posizione di atterraggio. In caso contrario si cadrà rovinosamente sulla neve, perdendo la gara.

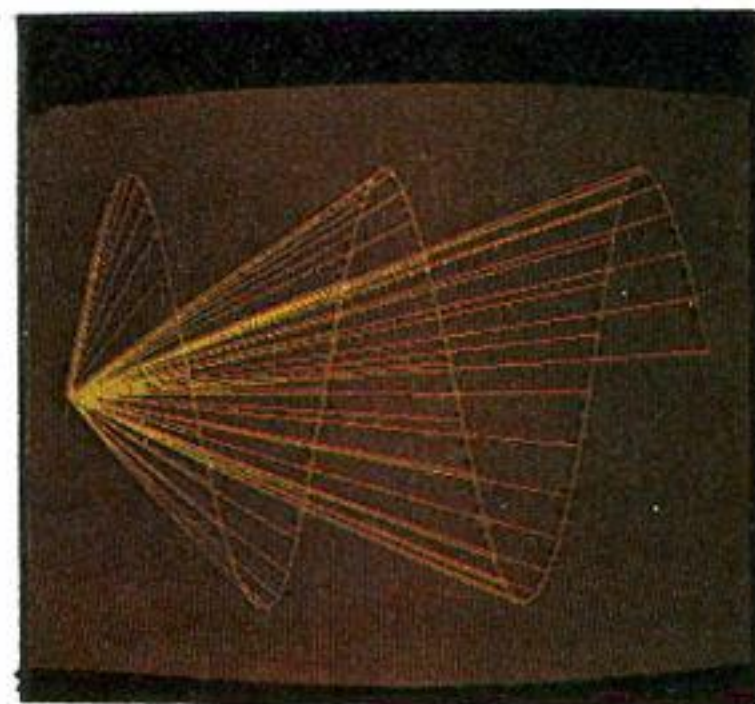
La distanza percorsa e lo "stile" del salto, determinano il punteggio, assegnato a fine gara.

5 - Speed Skating

E' una gara di velocità di pattinaggio sul ghiaccio, da disputarsi contro il computer o contro altri avversari. Per acquistare velocità è necessario muovere il joystick alternativamente a sinistra e a destra.

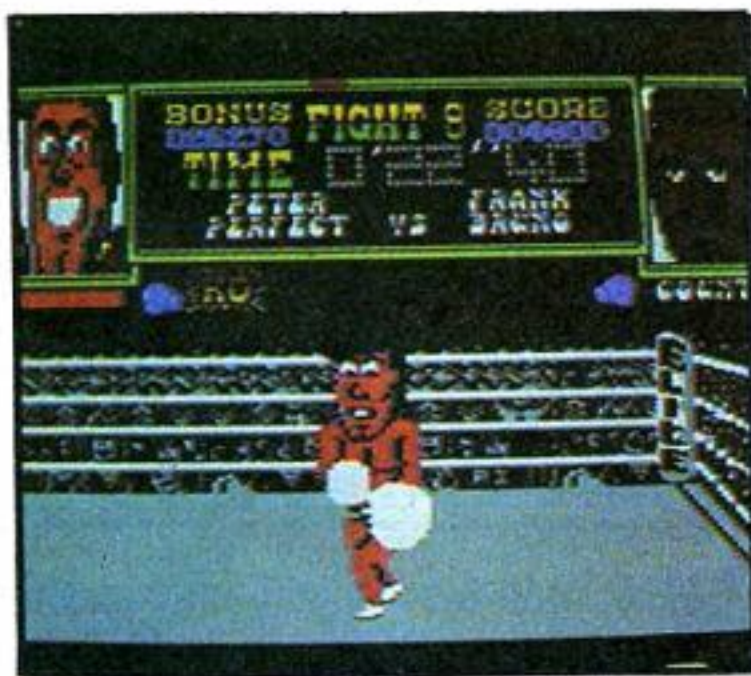
Lo schermo è diviso in due parti indipendenti, una per il giocatore e una per l'avversario, che indicano la velocità ed i metri percorsi. Vince, ovviamente, chi arriva prima, e, nel caso di un maggior numero di concorrenti, è valido il tempo minore.

6 - Free Skating

Questa gara è quasi identica al pattinaggio artistico, solo che le acrobazie sono libere, e non risulta necessario attenersi ad alcuna regola.

7 - Bobsled

Classica corsa su bob, tra le più veloci del gioco, in cui occorre, guidando il bolide a destra e a sinistra lungo la pista, cercare di non rovesciarsi o di uscire dai bordi. Il maggior punteggio si raggiunge portando a termine la corsa nel più breve tempo possibile.

Il termine del gioco

Alla fine delle gare verranno assegnate le medaglie d'oro, d'argento e di bronzo e, volendo, verranno registrati su supporto magnetico i record fatti dai concorrenti, che resteranno così disponibili al successivo caricamento del gioco.

Frank Brunòs boxing (Elite)

Parliamo ora di un altro prodotto fantastico, anch'esso di genere sportivo.

La software house è la Elite, famosa tra i commodoriani (e non) per aver sviluppato un parco di favolosi programmi.

F.B.Boxing fa parte appunto di questi.

Dopo averlo caricato, l'utilizzatore verrà deliziato da una bellissima e sofisticata grafica e da una perfetta esecuzione musicale della colonna sonora del film "Rocky", che sicuramente tutti conosceranno.

Rispetto ad altri programmi analoghi per C-64, questo rappresenta il massimo per ciò che riguarda grafica, suono, animazioni, opzioni, eccetera.

In seguito alla presentazione, si può decidere se vedere un dimostrativo oppure iniziare a giocare. In questo caso si affronterà il primo antagonista (in tutto sono nove, tra cui un Italiano dall'antipatico nome di "Ravioli Mafiosi": gli americani, si sa, affogano nei luoghi comuni).

Data la complessità del movimento, questo viene effettuato non con un solo joystick, ma addirittura con due contemporaneamente, uno per il movimento vero e proprio ed uno per colpire l'avversario. I possessori di un solo joystick dovranno pertanto affrettarsi ad acquistarne un altro (ma vi assicuriamo che ne vale la pena).

Nel tempo di tre minuti è necessario mettere ko l'avversario per tre volte, cercando, ovviamente, di non farci mettere ko a nostra volta.

Sullo schermo vengono indicate anche le energie rimaste a disposizione ed il tempo trascorso dall'inizio della partita.

Dopo aver messo ko la prima o la seconda volta un avversario, questo si rialza, cercando di colpirci con una terribile mossa segreta, difficilissima da evitare, che cambia per ogni avversario e che ci fa spesso finire al tappeto.

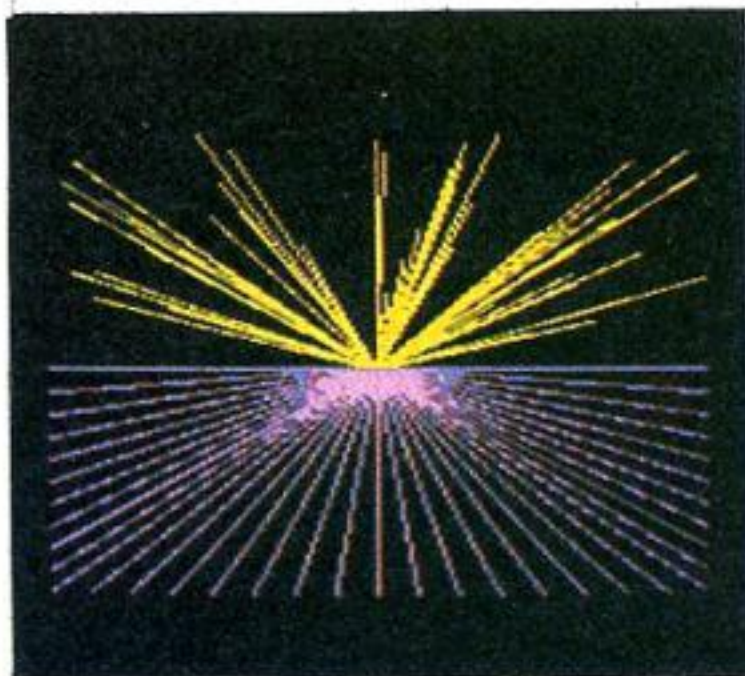
Se non riusciamo, entro i tre minuti, a mettere ko l'avversario o, se peggio, ci "stende" lui, la partita finisce.

In caso contrario viene chiesta la parola d'ordine per caricare il successivo boxer. Questa è generata casualmente dal computer, per cui ogni volta che carichiamo il gioco sarà necessario battere tutti i boxer in sequenza a partire dal primo, dato che non è possibile caricarli nell'ordine desiderato.

Continuando a sconfiggere avversari, la gara diventa sempre più difficile, rendendo quasi impossibile raggiungere il nono boxer.

Alla fine viene dato il punteggio totale di tutti gli avversari sconfitti.

Uno dei maggiori pregi di questo gioco è l'alta velocità e realismo di gioco, paragonabile agli analoghi giochi da bar.

Sarebbe stato meglio se ci fosse stata la possibilità di scegliere l'avversario con cui iniziare ma, anche così, il divertimento è assicurato.

I lettori che intendono collaborare sono pregati di telefonare nei pomeriggi di Martedì e Venerdì di ogni settimana (Tel. 02/8467348) allo scopo di valutare, pur se sommariamente, il valore dei propri programmi. Pertanto non inviate nastri o dischi o dattiloscritti, se non esplicitamente richiesti! Eviterete perdite di tempo (e di denaro).

Due games dalla Mastertronic

Non è vero che il Commodore 16 viene trascurato dalle case di software. Questi due giochi, apparsi di recente sul mercato, lo dimostrano.

Tutti frutti

Digitando RUN, dopo un pre-caricamento di 71 giri di nastro, compare una simpatica videata per alcuni secondi. In seguito il caricamento prosegue fino alla fine (totale: ben 168 giri).

Il gioco parte automaticamente accompagnato da una simpatica musichetta che sfrutta completamente le capacità sonore del computer. Se non si preme il tasto Fire del joystick vengono visualizzate l'una dopo l'altra le sei schermate che è possibile incontrare durante il gioco. Il numero di cinque vite a disposizione non sempre è sufficiente, specie le prime volte, a percorrere gli intricati labirinti lungo i quali dovete catturare ciliegie, evitando di farvi precipitare addosso enormi fragole.

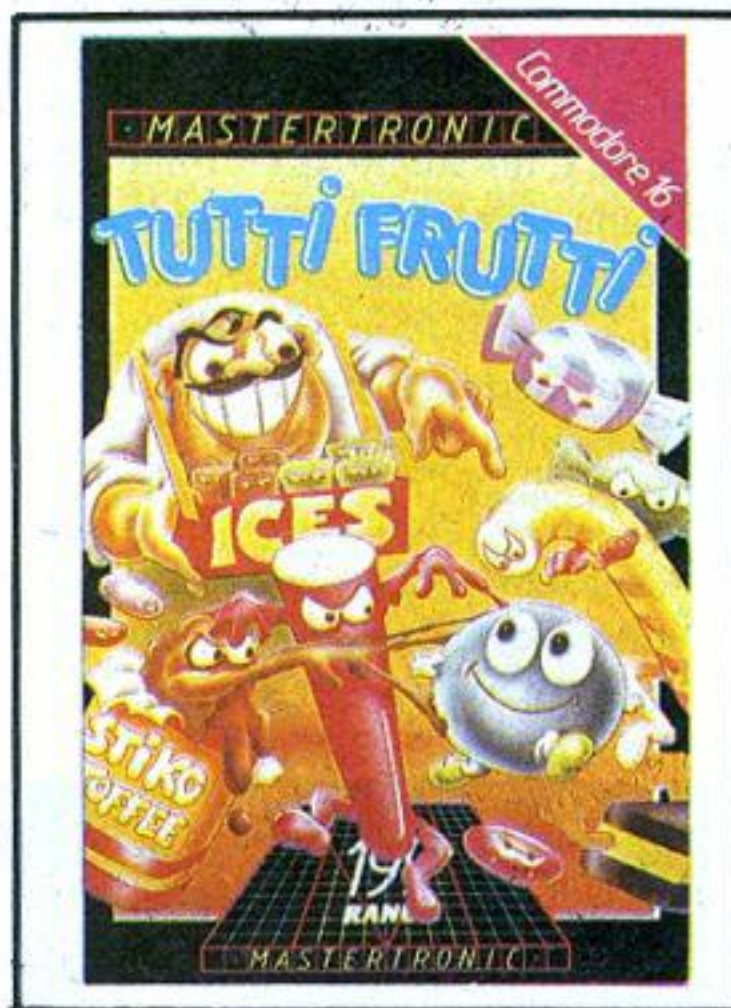
Nella prima schermata è sufficiente catturare tutte le ciliegie, evitando scontri con omini (in numero di cinque) che percorrono casualmente il vostro stesso percorso.

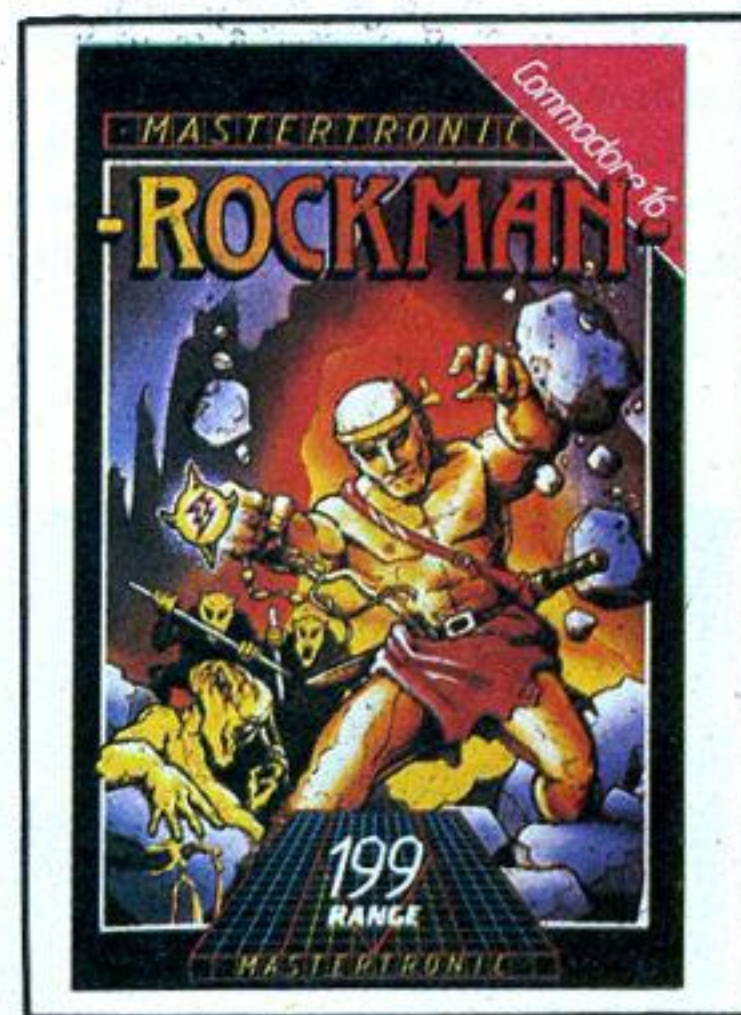
Catturata l'ultima ciliegia, si passa automaticamente al secondo schermo nel quale, dopo aver catturato le ciligie raffigurate, potete sparare, col tasto Fire, una pallina che percorre il labirinto rimbalzando con grande realismo lungo i bordi. E' necessario colpire, col la pallina, i cinque omini che, più agguerriti che mai, non si stancano di inseguirvi.

Passando agli schermi successivi il gioco si arricchisce di nuove regole: raccogliere ciliegie, spingere massi contro gli omini che vi inseguono, evitare di esser raggiunti.

Tutte le volte che il game termina, compare nuovamente la videata iniziale, il tempo impiegato dall'ultimo giocatore per condurre a termine la partita e gli ultimi otto punteggi più elevati.

Uno svantaggio è costituito dal fatto che è necessario il joystick per giocare.

Ci siamo però accorti che premedo il tasto F3 si dà inizio al gioco e, in seguito, si sparano le palline. Premendo il tasto Return l'omino si sposta in basso, col tasto di Lira sterlina (posta tra "=" e Shift), a sinistra, col tasto "4" a destra, e col tasto Ins/del in alto.

Rockman

Dopo un pre-caricamento di 17 giri, il programma viene interamente letto in 140 giri. In questo giocó non solo è possibile servisri del joystick e della tastiera,

ma è addirittura consentito scegliere i tasti da premere per guidare il nostro eroe lungo i labirinti del game.

Questo consiste nel catturare monete, diamanti ed altri oggetti preziosi disseminati lungo i numerosi labirinti. Catturandoli, però, si fanno cadere oggetti posti al di sopra di questi ed è quindi necessario esser lesti nell'allontanarsi. Anche in questo caso alcuni fantasmi tentano di impedirvi il proseguimento del gioco.

Da segnalare la musica che vi accompagna lungo tutto il gioco e alcuni effetti di animazione decisamente simpatici.

Nonostante il C-16, come è noto, non abbia la possibilità di generare sprite, questo gioco dimostra che ricorrendo ai caratteri programmabili è possibile visualizzare animazioni sufficientemente rapide ed originali. E' inutile dire, però, che è d'obbligo ricorrere alla tecnica del linguaggio macchina.

Una cartuccia carica di energia

di Alessandro de Simone

In commercio sono presenti numerosi programmi in grado di svolgere funzioni non previste dai costruttori dei computer. Tutti, però, devono essercaricati da nastro o disco prima di servirsi delle opportunità che offrono. Non sempre si ha la pazienza di effettuare tale procedura, un po' per pigrizia, un po' perchè, nel caso sia necessario spegnere la macchina. bisognerebbe ricominciare tutto daccapo.

Si apprezzano volentieri, pertanto, cartucce da inserire sul retro del C-64 che entrano immediatamente in funzione non appena si accende l'apparecchio.

Ne abbiamo provata una (ed inserito permanentemente sulle nostre macchine...) ed esattamente la TURBO 50 distribuita dalla Mastertronic (Viale Aguggiari, 63/A - 21100 Varese - tel. 0332/238898).

Questa cartuccia non occupa memoria RAM riservata al Basic, che dispone, quindi, dei soliti 38911 bytes liberi. Utilizza, però, una parte della RAM posizionata a partire dall'indirizzo 49152. Per caricare i programmi in Linguaggio Macchina (L.M.) che occupano tali locazioni è quindi necessario disabilitare la cartuccia via software senza, cioè, prendersi il disturbo di staccarla fisicamente dal retro. Con il pulsante di Reset presente sulla stessa cartuccia, infatti, il Commodore 64 si comporta come se la cartridge fosse disinserita.

Ma bando ai preamboli, e passiamo alla descrizione, pur se sommaria per ragioni di spazio, delle nuove potenzialità offerte da TURBO 50.

Come è bello accendere il C-64 ed avere immediatamente disponibili tanti comandi BASIC in più, oltre alla possibilità di aumentare la velocità di caricamento da nastro o disco...

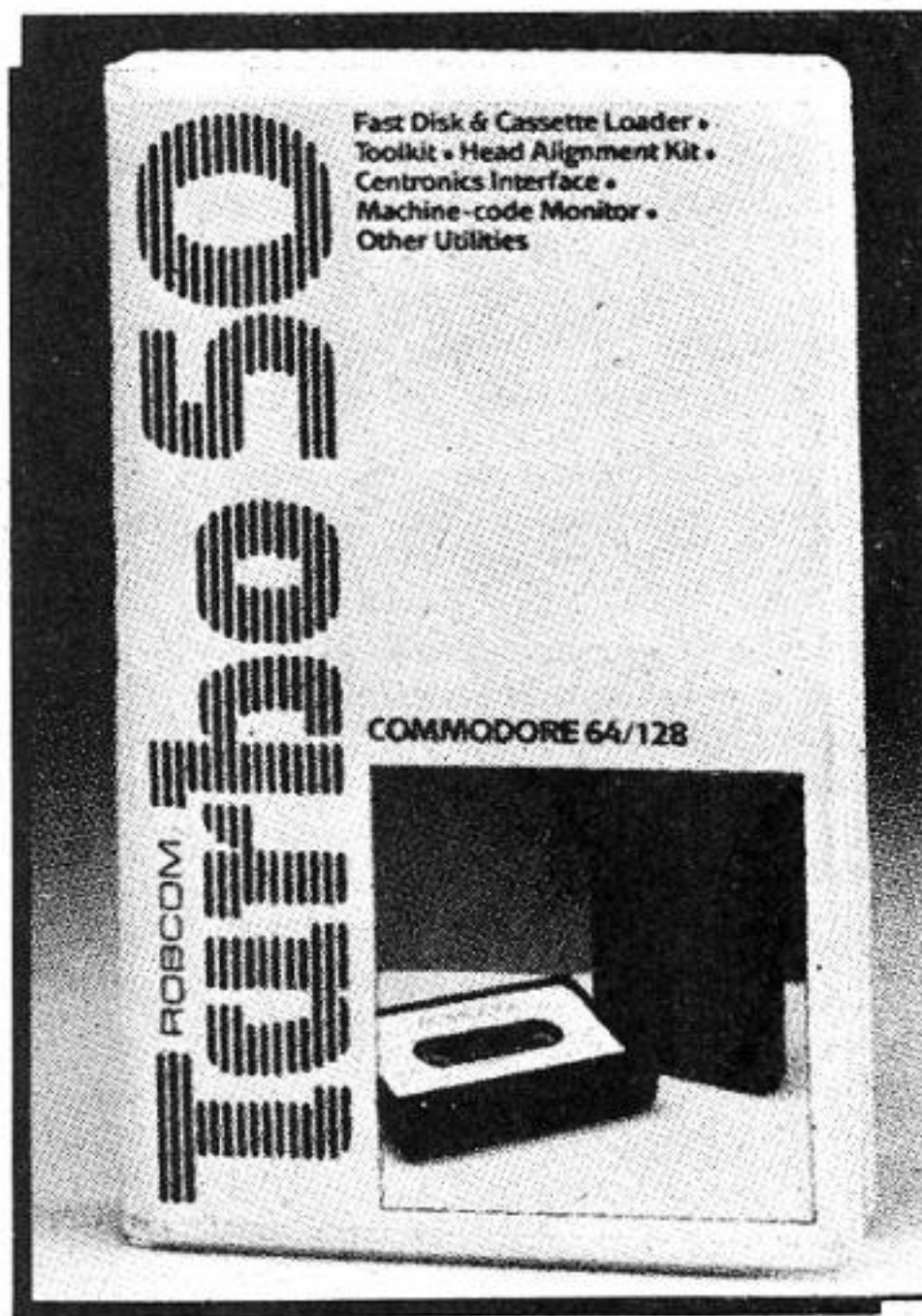
Comandi Basic aggiuntivi

AUTO N,M
Genera automaticamente la numerazione di linee di un programma a partire da N con incremento M.

DEC N
Traduce il numero esadecimale N in decimale.

DEL N-M
Cancella le linee Basic da N a M incluse.

FIND ELEMENT
Ricerca e visualizza tutte le righe del programma Basic in cui è presente l'elemento cercato ELEMENT (istruzione, comando, stringa, variabile, ecc.).

HELP
In caso di errore visualizza l'esatta istruzione Basic che ha generato l'errore stesso.

HEX N
Traduce in esadecimale il numero decimale N.

KEY N
Abilita o disabilita le funzioni preprogrammate coi tasti funzione f1-f8.

OLD
Fa "resuscitare" il programma Basic cancellato accidentalmente col comando NEW, oppure QUIT o RESET (vedi dopo).

PLIST
Visualizza pagina dopo pagina il programma Basic in memoria in modo tale, cioè, da bloccare ogni 25 righe la funzione di scrolling che farebbe altrimenti uscire dallo schermo le righe del listato.

QUIT
Elimina, via software, la cartuccia ed il C-64 "pensa" che non ci sia più. Se il comando è stato dato per errore, la si può attivare nuovamente col comando SYS 49179.

RENUM N,M
Renumera il programma Basic a partire da N con intervallo M.

REP
Abilita la funzione di autorepeat su tutti i tasti.

RESET
Cancella totalmente ciò che è in memoria, disabilita la cartuccia e presenta C-64 come se questo fosse stato appena accesso.

SET N,M
Colora il bordo col colore N ed il fondo con M.

SHOW
Visualizza su schermo tutti i comandi disponibili con la cartuccia.

TRACE N
Evidenzia su schermo, durante l'elaborazione di un programma, le linee che vengono a mano a mano eseguite.

Comandi per file su cassetta

COMP
Verifica la corretta registrazione di un programma salvato col comando PUT (vedi dopo).

GET "NOME"
Carica 10 volte più velocemente un programma registrato precedentemente col comando PUT.

MERGE
Esegue una fusione tra due pro-grammi.

PUT "NOME"
Registra il programma Basic con una velocità 10 volte più elevata del consueto.

Comandi per files su disco

DLOAD"NOME"
Carica ad una velocità 5 volte superiore al consueto un programma Basic.

DMERGE "NOME"
Fonde il programma NOME presente sul dischetto col programma presente in memoria.

DSAVE"NOME"
Registra ad una maggiore velocità il programma presente in memoria.

DVERIFY"NOME"
Verifica il programma appena regi-strato.

↑ (freccia in alto)
Visualizza la directory senza cancellare il programma Basic presente in memoria.

(punto)
Indica quale errore è stato commesso nel caso il drive lampeggi.

.N .S .R .V .U
Questi comandi, preceduti dal carattere di punto (.), eseguono, rispettivamente, la formattazione di un disco; la cancellazione di un file; il cambio di un nome; il validate; la disabilitazione del drive.

/ "NOME"
Digitando il nome del programma, preceduto dal carattere di barra (/), il programma viene caricato a gran velocità a partire dalla locazione in cui era presente al momento della registrazione. Corrisponde, cioè, al comando: LOAD"NOME",8,1

FLOPPY N
Assegna il numero 8 oppure 9 al floppy collegato in quel momento col C-64.

Comandi per copiare files

COPY"NOME",N,M
Copia il file "NOME" (lungo fino ad un massimo di 186 blocchi!) dalla periferica N (registratore t2; drive t8) alla periferica M. E' ovvio che il caricamento può avvenire anche ad alta velocità e che il programma effettua una pausa per consentire l'ventuale cambio del nastro e/o disco.

RECOPY
Consente di effettuare un'altra copia del programma o file appena compiuta col comando COPY.

Comandi per l'uso dell'interfaccia Centronics

CENT A,B
Attiva o disabilita l'interfaccia Centronics e la predispone, o meno, in modo Listing Mode.

COLUMN Q
Determina il numero Q di caratteri da stampare per ciascuna riga.

Allineamento della testina del registratore

Nella confezione è presente una speciale cassetta che, inserita nel registratore ed utilizzata col comando GET, consente la perfetta regolazione dell'azimuth grazie alla visualizzazione di strisce colorate che appaiono sullo schermo del TV.

Comandi monitor

Un programma per la gestione del Linguaggio Macchina è presente nella cartuccia e la sua attivazione è possibile digitando il comando MON.

La notevole particolarità risiede nel fatto che questo programma, lungo 4Kbytes, è rilocabile nel senso che è possibile allocarlo nella parte di memoria RAM che più aggrada. Chi lavora in L.M. sa bene i notevoli limiti imposti da altri programmi similari che, pur se validissimi, occupano una ben precisa parte di memoria RAM impedendo, di fatto, di allocarvi programmi particolari. Con TURBO 50, al contrario, potetete scrivere listati in L.M. in qualsiasi parte della memoria RAM.

I comandi disponibili, sui quali saremo ancora più succinti, sono:

MEMTOP Fissa il punto più alto di memoria utilizzabile.

ASSEMBLE Consente di scrivere il programma in Assembler.

BREAKPOINT Consente di inserire all'interno di un programma un punto di Break.

COMPARE MEMORY Confronta tra loro due segmenti di memoria.

DISASSEMBLE Disassembla una zona di memoria.

EDITOR IN SCREEN Esamina una zona di memoria.

FILL MEMORY Riempe con un numero esadecimale una porzione di memoria.

GO Corrisponde al RUN.

HUNT MEMORY Ricerca una sequenza di bytes all'interno di una zona di memoria.

INTERROGATE MEMORY Visualizza una zona a blocchi di otto byte per rigo con la corrispondente "traduzione" ASCII.

NEWLOCATER riloca gli indirizzi assoluti di un programma L.M. Analogamente la funzione ZIP riloca tabelle.

Altre funzioni e comandi, di facile interpretazione, sono: Jump, Subroutine, Load, Printer Output, Register Display, Save, Transfer Memory, Video Screen, Exit.

Sono inoltre disponibili i comodi comandi per effetture somme e sottrazioni in esadecimale e conversioni di numeri.

Conclusioni

Sperando di non aver dimenticato nessuna delle potenzialità offerte da TURBO 50, concludiamo asserendo che il prezzo (L. 125.000) può sembrare non del tutto contenuto. Considerando, però, i vantaggi offerti dalla cartuccia, noi pensiamo che questa si ripaghi da sè...

SCAMBIATEVI LE LISTE *o promuovete un Club*

Marco Scorta - Via S. Francesco 13 - 20091 Bresso (MI) - tel. 02/6100061.

Alessandro D'Amore - Via Umberto 16 - 73048 Nardo (LE).

Andrea Simoni - Via Isonzo 16 - Castelfranco E. - tel. 059/937205.

Francesco Casumano - Via C. Matteotti C.le 5 35 - 92013 Menfi - tel. 0925/71309.

Giovanni Pasquariello - Via Torre S. Susanna 129 B - 72022 Latiano (BR) - tel. 729291.

Marco Barili - Via Piave 4 - 26026 Pizzighettone (CR) - tel. 0372/743052.

Pietro Pulier - Via Vico II Fontepliniano 12 - 74024 Manduria - tel. 099/6791211.

Maximiliano De Martis - Via Delfico 24/3 - 20155 Milano - tel. 02/3493615.

Niccolò Vecciarelli - P.zza Vinci 29 - 00139 Roma - tel. 06/8109780.

Mirco Zanca - Via Enrico Fermi 3 - 46035 Ostiglia (MN) - tel. 0386/31657.

Romano Piacentini - Via Vol del Sangue 171/A - 20099 Sesto S. Giovanni (MI) - tel. 02/2428447.

Mario Gatti - Via Sebenia 10 - 25040 Clusane d'Iseo (BS) - tel. 030/989286.

Piero Graziani - Via Delle Torri 72 - 50019 Sesto Fiorentino (FI).

Marco e Giuliano Quaranta - Via Malta 9 - 80055 Portici (NA) - tel. 081/484584.

Raffele Rondoni - Via Del Babuino 151 - 00187 Roma - tel. 06/6798610.

Walter Bellese - Via Borgovecchio 12 - 30027 S. Donà di Piave (VE).

Andrea Magris - Via Daniele Manin 112 - 31015 Conegliano (TS) - tel. 0438/63313.

Giuseppe Chiocci - Via Savelli della Porta 8 - 06024 Gubbio (PG) - tel. 075/9271549.

Gaetano Di Bello - Via Luigi Caterino 83 - 81036 S. Cipriano D'Aversa (CE) - tel. 081/8921268.

Massimo Pedriali - Via Gondini 44 - 44100 Ferrara - tel. 0532/903354.

Stefano Sgadò - Via Don Minzoni 2/A - 54033 Carrara (MS) - tel. 0585/72127.

Franco Silvestro - P.zza S. Martino 9 - 40126 Bologna - tel. 227074.

Ferdinando Bertazzini - Via Curiel 19 - 40013 Castelmaggiore (BO) - tel. 051/712580.

Salvatore Cerchione - Via Ribera 5 - 80128 Napoli - tel. 081/649108.

Lorenzo De Angelis - Via Braccianese 8 - 00059 Tolfa - tel. 0766/92615.

Edoardo Tratta - Via Lazio 3 - 80016 marano (NA).

David Malizia - Via L.T. Inventori 60 - 00146 Roma - tel. 5561379.

Antonio Villani - P.zza Ischia 2 - 00141 Roma - tel. 06/8174100.

Stefano Scaini - Via C. Max 15 - 43100 Parma - tel. 0521/44354.

Sandro Matali - Via XIV Luglio 36A - 50019 Sesto F.no (FI) - tel. 055/4481640.

Mario Berardo - Via Vittorio Emanuele 265 - 12042 Bra (CN) - tel. 0172/421703.

Demetrio Bellardini - Via Simone Ghini 32 - 00133 Roma - tel. 06/2692609.

Giovanni Bonatelli - Via Lussino 31 - 37139 Verona - tel. 582719.

Gianluca Riccardi/Costantino Mastracco - Via P.D. Passionista 3B - 03024 Ceprano (Fr) Tel. 0775/950102

Bruno Castelnuovo - Via S. Antonio 3 - 23020 Montagna in Valtellina (So) - Tel. 0342/380234

Daniele Peruzzi - Via J. Cook 8 - 52100 Arezzo Tel. 0575/911306

Eugenio Ferla - Via P. Cominio 56 - 00175 Roma - Tel. 06/765535

Bernadotti Flavio - Via Trento 10 - 15040 Montecastello (Al)

Fabio Bellassai - Via F.lli Bandiera 13 - 95100 Catania - Tel. 415353

Sebastiano Caramagno - Contrada Cipollazzo 96011 Augusta - Tel. 0931/993333

Erminio Dessì - V.le del Lavoro 20 - 87012 Castrovillari (Cs) - Tel. 0981/46393

Riccardo Donadon - Via Postumia 22 - 31055 Quinto di Treviso - Tel. 0422/20894

Antonio Conticello - Via Galli della Mantica 28 - 00122 Ostia (Rm) - Tel. 06/5603759

Michele Cervelli - Via Fiorentina 43 - 56025 Pontedera (Pi) - Tel. 0587/56046

Enzo Landro - Via delle Terme 97 - 95024 Acireale (Ct) - Tel. 095/608294

Riccardo Badenghini - Via Pianezza 153 - 10151 Torino - Te. 011/734940

Paolo Pagani - Via Guillenzani 5 - 41100 Modena - Tel. 059/350964

Simone Merlini - Via Baglioni 24 - 06100 Perugia - Tel. 075/61580

Oscar Bresciani - Via Veneto 24 - 20031 Cesano Maderno (Mi)

Luca Brugnera - Via Comunale 46 - 31046 Oderzo (Tv) - Tel. 0422/717574

Loredana Meccariello - Via Goti 97 - 42019 Scandiano - Tel. 0522/855130

Claudio Dammicco - Via Squillace 64 - 00178 Roma - Tel. 06/7992123

Gerardo Marra - Via P. Baratta 91/A - 84091 Battipaglia (Sa) - Tel. 0828/23130

Francesco Saracino - Via E. Saracino 25 - 70032 Bitonto (Ba) - Tel. 617600

Marco Cavedoni - Via G. Pascoli 95 - 41058 Vignola (Mo) - Tel. 059/771734

Salvatore Spanedda - C.so Umberto - 09073 Gugliera - Tel. 0785/39857

Sandro Garzelli - Via G. Amendola 17 - 50053 Empoli (Fi) - Tel. 0571/78361

Frabrizio Rizzi - Via Castello 3060B - 30122 Venezia - Tel. 041/22883

Marco Mori - Via F. Crispi 13 - 52100 Arezzo - Tel. 0575/22013

Primo Valmori - Via della Resistenza 18 - 48020 Ca' di Lugo (Ra)

Roberto Pallotti - Via Audinot 35 - 40134 Bologna - Tel. 051/421956

Umberto Morra di Cella - Via Gen. Perotti 23 10091 Alpignano (To) - Tel. 011/9677679

Massimiliano Gucciardi - Via Brecce Bianche 80 - 60131 Ancona - Tel. 071/862641

Antonio Santagata - Via Valbruna 16 - 33100 Udine - Tel. 43456

Igor Princic - V.le XX Settembre 35 - 34137 Gorizia - Tel. 0481/89200

Massimo Fortin - Via Rovetta 6 - 20127 Milano - Tel. 2852865

Poalo Solaro - P.zza Medaglie d'Oro 13 - 14100 Asti - Tel. 0141/51973

Roberto Michelotti - Via G.B. Gaulli 11/23 - 16143 Genova

Maurizio Belloni - Via Legionari Polonia 33 - 24100 Bergamo - Tel. 035/262083

Giulio Manni - Via Gallipoli 73 - 73055 Racale (Le) - Tel. 0833/982551

Roberto Facco - Via Varesina 181 - 22100 Como - Tel. 031/505896

Andrea Gamannossi - Via T. Cortesi 19 - 50047 Prato (Fi)

Elisa Leoni - P.zza Giotto 20 - 52100 Arezzo - Tel. 0575/20606

Edoardo Lampis - Via Cadorna 21 - 50129 Firenze - Tel. 055/494648

Stefano Sampaolesi - Via A. Moro 15 - 63021 Amamdola (Ap) - Tel. 0736/97462

Albano Filiaci - Via B. Miriam 61 - 63035 Offida (Ap) - Tel. 0736/80144

Marco Milesi - Via Dante 55 - 24032 Calolziocorte (Bg) - Tel. 0341/645507

Massimiliano Mattioli - Via G. Paganelli 10 - 47100 Forlì - Tel. 0543/51261

Michele Melchionda - Via Tagliamento II Traversa 6 - 83100 Avellino - Tel. 0825/38091

Roberto Oberto - Via Biscaretti 7 - 10025 Pino Totinese (To) - Tel. 011/842357

Roberto Guida - P.zza Ernesto Cesaro 66 - 80058 Torre Annunziata - Tel. 081/8616561

Angelo Germinario - Via F. Fusco 36 - 95100 Catania - Tel. 095/437234

Andrea Franci - Via A. de Gasperi 34 - 52100 Arezzo - Tel. 0575/33540

Giorgio Franchi - C.so Quintino Sella 87 - 10132 Torino - Tel. 011/832875

Marco Maggi - Via Serlio 8/2 - 20139 Milano

Roberto Boselli/Andrea Padovani - Via Gonzales 4 - 20139 Milano - Tel. 02/567465/5692965

Sergio Rivoletti - Via Tevere - 63017 Porto San Girogio - Tel. 0734/4419

Marco Arcari - Via Walter Muttini 14 - 54036 Massa Carrara (Ms)

Claudio Luciani - Via J. Gagarin 34 - 63020 Piane di Falerone

Corrado Ferrari - Via Europa 40 - 20097 S. Donato Milanese (Mi) - Tel. 02/5274458

Luigi Buzzi - Via A. Grandi 3 - 12011 Borgo S.D. - Tel. 0171/751248

Poalo Rivieccio - Via Montebello 115 - 57100 Livorno - Tel. 0586/807043

INVIARE TUTTA LA PAGINA ANCHE SE SI UTILIZZA UNA SOLA SCHEDA

Nome .... Cognome ....

Via .... N° .... CAP. .... Città ....

Telefono .... Orario ....

Registrate il mio abbonamento annuale a Commodore Computer Club.

☐ Ho versato oggi stesso il canone di L. 35.000 a mezzo c/c postale n°37952207 intestato a:
Systems Editoriale - V.le Famagosta, 75 - 20142 Milano

☐ Ho inviato oggi stesso assegno bancario n..
per l'importo di L. 35.000 intestato a Systems Editoriale

Si prega di scrivere il proprio nome e l'indirizzo completo in modo chiaro e leggibile. Inviare la fotocopia del bollettino di c/c postale.

Considerando che i numeri 1, 2 e 7 sono esauriti, vogliate inviarmi i numeri arretrati .... al prezzo di L. 5.000 cadauno per richieste fino a 4 numeri, o di L. 4.000 cadauno per richieste oltre i 4 numeri arretrati, e perciò per un totale di L.............. Sono a conoscenza che i fascioli suddetti non saranno inviati in contrassegno e, pertanto, ho provveduto oggi stesso a versare il canone di L.................. a mezzo c/c postale n. 37952207 intestato a:
Systems Editoriale - V.le Famagosta, 75 - 20142 Milano

STATISTICA

Non posseggo un computer ☐

Posseggo un C64	si ☐	... no ☐
Posseggo un VIC 20	si ☐	... no ☐
Posseggo un Commodore Plus 14	si ☐	... no ☐
Posseggo un Commodore Plus 16	si ☐	... no ☐
Posseggo un registratore dedicato	si ☐	... no ☐
Posseggo un drive 1541	si ☐	... no ☐
Posseggo una stampante	si ☐	... no ☐
Posseggo un monitor	si ☐	... no ☐

COLLABORAZIONE

A titolo di prova vi invio un articolo e la cassetta........................disco..................................
col programma che intendo proporre per la pubblicazione di cui garantisco l'originalità.

DOMANDA/RISPOSTA ....

....

....

....

....

....

RICHIESTA ARGOMENTI

Mi farebbe piacere che Commodore Computer Club parlasse più spesso dei seguenti argomenti:

1/ ....

2/ ....

3/ ....

4/ ....

GIUDIZIO SUI PROGRAMMI DI QUESTO NUMERO

Ho assegnato un voto da 0 a 10 ai programmi che indico di seguito:

A/ .... Voto ....

B/ .... Voto ....

C/ .... Voto ....

D/ .... Voto ....

PICCOLI ANNUNCI

....

....

....

....

....

CERCO/OFFRO CONSULENZA

....

....

....

....

....

INVIARE IN BUSTA CHIUSA E AFFRANCANDO SECONDO LE TARIFFE VIGENTI A:

COMMODORE COMPUTER CLUB

V.le Famagosta, 75
20142 Milano

INVIARE TUTTA LA PAGINA ANCHE SE SI UTILIZZA UNA SOLA SCHEDA

Nome .... Cognome ....

Via .... N° .... CAP. .... Città ....

Telefono .... Orario ....

systems

systems